Multi-target tracking and performance evaluation on videos

by

Fabio Poiesi

Bachelor in Telecommunication Engineering

Master in Telecommunication Engineering

A dissertation submitted to

The School of Electronic Engineering and Computer Science

in partial fulfilment of the requirements for the Degree of

Doctor of Philosophy

in the subject of

Electronic Engineering

Queen Mary University of London

Mile End Road

E1 4NS, London, UK

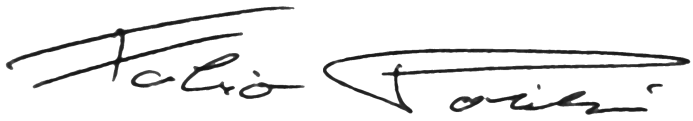
December 2013

I, Fabio Poiesi, confirm that the research included within this thesis is my own work, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature: 

Date: 17th March 2014

ai miei genitori

Thesis supervisor
Professor Andrea Cavallaro

Author
Fabio Poiesi

Multi-target tracking and performance evaluation on videos

Abstract

Multi-target tracking is the process that allows the extraction of object motion patterns of interest from a scene. Motion patterns are often described through metadata representing object locations and shape information. In the first part of this thesis we discuss the state-of-the-art methods aimed at accomplishing this task on monocular views and also analyse the methods for evaluating their performance. The second part of the thesis describes our research contribution to these topics.

We begin presenting a method for multi-target tracking based on track-before-detect (MT-TBD) formulated as a particle filter. The novelty involves the inclusion of the target identity (ID) into the particle state, which enables the algorithm to deal with an unknown and unlimited number of targets. We propose a probabilistic model of particle birth and death based on Markov Random Fields. This model allows us to overcome the problem of the mixing of IDs of close targets.

We then propose three evaluation measures that take into account target-size variations, combine accuracy and cardinality errors, quantify long-term tracking accuracy at different accuracy levels, and evaluate ID changes relative to the duration of the track in which they occur. This set of measures does not require pre-setting of parameters and allows one to holistically evaluate tracking performance in an application-independent manner.

Lastly, we present a framework for multi-target localisation applied on scenes with a high density of compact objects. Candidate target locations are initially generated by extracting object features from intensity maps using an iterative method based on a gradient-climbing technique and an isocontour slicing approach. A graph-based data association method for multi-target tracking is then applied to link valid candidate target locations over time and to discard those which are spurious. This method can deal with point targets having indistinguishable appearance and unpredictable motion.

MT-TBD is evaluated and compared with state-of-the-art methods on real-world surveillance

datasets (static and moving cameras) by using the proposed evaluation measures. In the case of online applications the inclusion of the ID in the particle state is effective, but it does not allow the proposed tracker to outperform offline trackers. The proposed measures are compared with existing measures for multi-target tracking and it is shown that the proposed ones comparatively maintain a reliable evaluation of the performance without prior knowledge about the application. The tracking of point targets in high-density scenes is evaluated on datasets containing insects and compared with MT-TBD and alternative multi-target trackers. The proposed solutions achieved the best results, especially in terms of ID maintenance on the targets.

Contents

Acknowledgements

I want to primarily thank my Parents. Their moral support, motivational advice and constant presence helped me to get to the point where I am now. Grazie per tutto.

I want to thank my supervisor Professor Andrea Cavallaro for his patience in giving me extremely useful and unique technical advice during these four quick years.

I want to thank all my friends who worked with me for the unforgettable moments spent together inside and outside the lab.

This work was supported by the EU, under the FP7 project APIDIS (ICT-216023) and the Artemis JU and TSB as part of the COPCAMS project (332913).

Published work

Journal papers

[J1] T. Nawaz, F. Poiesi and A. Cavallaro. Measures of effective video tracking. *IEEE Trans. on Image Processing*, vol. 23, no. 1, pp. 376-388, Jan. 2014.

[J2] F. Poiesi, R. Mazzon and A. Cavallaro. Multi-target tracking on confidence maps: an application to people tracking. *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1257-1272, Oct. 2013.

Book chapter

[B1] F. Poiesi and A. Cavallaro. Multi-target tracking in video. *Academic Press Library in Signal Processing: Volume 4*, (Ed. S. Theodoridis), Elsevier, Sep. 2013.

Conference papers

[C1] R. Mazzon, F. Poiesi and A. Cavallaro. Detection and tracking of groups in crowd. *IEEE Proc. of Advance Video and Signal Based Surveillance*, Krakow, Poland, Aug. 2013, pp. 202-207.

Electronic preprints are available on the at:
http://www.eecs.qmul.ac.uk/staffinfo/andrea/publications.html

Glossary of abbreviations

ADA	Application-Dependent Assignment-based evaluation	29
AER	Accuracy Error Rate	73
BRIEF	Binary Robust Independent Elementary Features	16
CDT	Correct Detected Track	33
CER	Cardinality Error Rate	73
CRFBT	Conditional Random Field Based Tracker	75
DP-NMS	Dynamic Programming-Non-Maxima Suppression based tracker	75
EM	Expectation-Maximisation	28
EOM	Explicit Occlusion Model	26
F	F-score	29
FAR	False Alarm Rate	30
FAT	False Alarm Track	33
FN	False Negative	29
FP	False Positive	29
GCD	Gradient-Climbing based Detector	101
GGB	Greed Graph Based	90
GMM	Gaussian Mixture Model	39
HA	Hungarian Algorithm	97
HIM	Hierarchical-Isocontour based Morphology	101
HMM	Hidden Markov Model	20
HOG	Histogram of Oriented Gradient	15
ID	Identity	iv
IDC	Identity Changes	34
IDS	Identity Switches	61
IDSR	Identity Switch Rates	102

KF	Kalman Filter	107
KLT	Kanade-Lucas-Tomasi	19
LBP	Local Binary Patterns	16
M-H	Metropolis-Hastings	93
MAP	Maximum A Posteriori	23
MCMC	Monte Carlo Markov Chain	17
MCMCDA	Monte Carlo Markov Chain Data Association	62
MELT	Multiple Extended-target Lost-Track	37
METE	Multiple Extended-target Tracking Error	37
MHT	Multiple Hypothesis Tracking	18
MODA	Multiple Object Detection Accuracy	33
MOTA	Multiple Object Tracking Accuracy	34
MOTP	Multiple Object Tracking Precision	33
MRF	Markov Random Field	39
MS	Mean-Shift	45
MSER	Maximally Stable Extremal Regions	15
MT-TBD	Multi-Target Track-Before-Detect	iv
N-MODA	Normalised Multiple Object Detection Accuracy	34
NIDC	Normalised ID Changes	37
NMS	Non Maxima Suppression	2
OLDAM	On-line Learned Discriminative Appearance Models	25
OSPA	Optimal Sub-Pattern Assignment	30
OTE	Object Tracking Error	30
P	Precision	29
PAP	Point-based Assignment and Position-based evaluation	29
PCA	Principal Component Analysis	25
PHD	Probability Hypothesis Density	27
PP	Postprocessing	101
R	Recall	29
RAP	Region-based Assignment and Position-based evaluation	29
RAS	Region-based Assignment and Size-based evaluation	29

RFS	Random Finite Sets	22
RGB	Red Green Blue	15
SIFT	Scale-Invariant Feature Transform	16
SNR	Signal-to-Noise-Ratio	2
SVM	Support Vector Machines	2
TBD	Track-Before-Detect	iv
TDF	Track Detection Failure	33
TDR	Track Detection Rate	30
TF	Track Fragmentation	30
TP	True Positive	29
TRDR	Tracker Detection Rate	30

Glossary of symbols

$\mathcal{V}$	video sequence	3
v_k	k^{th} video frame of the video sequence	3
K	number of video frames of $\mathcal{V}$	3
$\mathcal{Z}_k$	set of features/detections generated by a detector at k	3
B	total number of features computed within $\mathcal{V}$	3
$\mathbf{z}_{b,k}$	b^{th} feature/detection at k	3
$x_{b,k}$	horizontal position of the b^{th} feature/detection	3
$y_{b,k}$	vertical position of the b^{th} feature/detection	3
$S_{b,k}$	shape or scale of the b^{th} feature/detection	3
$\iota_{b,k}$	confidence value of the b^{th} feature/detection	3
$\mathcal{T}$	set of estimated tracks in $\mathcal{V}$	4
$\mathcal{T}_k$	set of estimated tracks at k	4
T_a	track of the a^{th} target	4
$\mathsf{T}_{a,k}$	track of the a^{th} target up to k	4
A	total number of tracks computed within $\mathcal{V}$	4
u_k	number of tracks at k	4
$\mathbf{x}_k$	generic state of a tracked target at k	40
$\mathfrak{X}$	state space defining the target state	40
x_k	estimated horizontal position of a target at k	40
y_k	estimated vertical position of a target at k	40
$\dot{x}_k$	estimated horizontal velocity of a target at k	40
$\dot{y}_k$	estimated vertical velocity of a target at k	40
I_k	estimated confidence value of a target at k	40
$F_{\mathbf{x}}$	prior evolution model of the target state	40
$\hat{\mathbf{x}}_{a,k}$	estimated state of the a^{th} extended target of $\mathsf{T}_{a,k}$ at k	4
$\hat{\mathbf{x}}'_{a,k}$	estimated state of the a^{th} point target of $\mathsf{T}_{a,k}$ at k	4

ξ	generic estimated target identity	41
ξ_a	estimated identity of a^{th} target	4
${}^{g}\mathcal{T}$	ground-truth tracks of a video	4
${}^{g}\mathcal{T}_k$	ground-truth tracks at k	31
${}^{g}\mathsf{T}_d$	ground-truth track of the d^{th} target	4
${}^{g}\hat{\mathbf{x}}_{d,k}$	ground-truth state of the d^{th} extended target of $\mathsf{T}_{d,k}$ at k	4
${}^{g}\hat{\mathbf{x}}'_{d,k}$	ground-truth state of the d^{th} point target of $\mathsf{T}_{d,k}$ at k	4
${}^{g}u_k$	number of ground-truth tracks at k	4
${}^{g}\xi_d$	ground-truth identity of the d^{th} target	4
τ_{TP}	overlap threshold to consider an association as a True Positive	30
$h_k^{(i,j)}$	2D spread function of the estimated positions of a generic target at pixel position (i,j) at k	40
$p(\cdot)$	probability density function	41
$\mathcal{L}_k$	set of target identities defined as random variables at k	41
L_ξ	random variable representing the target identity ξ	41
Ξ_k	set of identities at k	41
$g_{ID}(\cdot)$	function that (i) maintains target identities, (ii) assigns new identities to appearing targets and (iii) removes the identities of targets that have disappeared	41
$\mathfrak{N}(\xi)$	neighbouring identities to L_ξ	42
$\mathfrak{N}(\xi^n)$	neighbouring particles to the n^{th} particle	46
$\mathbf{x}_k^n$	state of the n^{th} particle at k	43
ξ^n	identity of the n^{th} particle	43
w_k^n	importance weight of the n^{th} particle at k	43
Q_k	set of existing particles at k	43
J_k	set of new-born particles at k	43
N	total number of particles ($N = Q_k + J_k$)	43
$q_k(\cdot)$	proposal distribution to propagate particles at k	43
$\ell(\cdot)$	likelihood function	45
λ_Ψ	size of the cluster	45
Ψ_k	set of clusters at k	45

ψ_r	r^{th} cluster of Ψ_k	45
$\mathcal{R}_k$	number of clusters of Ψ_k	45
$V_{\mathfrak{N}(\xi)}$	potential function defined for the neighbourhood $\mathfrak{N}(\xi)$	46
$\varkappa_k^n$	quantifies of the agreement of the identities at k	47
α_1	regulates the strength of the agreement $\varkappa_k^n$ defined in $V'_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$	47
α_2	regulates the decreasing trend of $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$	47
δ_k^n	Dirac function that indicates if n is a new-born particle or not at k	47
$\mathcal{X}_r$	set of particle locations and identities belonging to ψ_r	49
θ_ξ	mean position of particles with identity ξ	49
Θ_r	set of mean positions within ψ_r	49
$\mathfrak{T}$	set of short-term tracks of $\mathcal{V}$	97
$\mathfrak{t}_l$	single short-term track belonging to $\mathfrak{T}$	97
L	number of short-term tracks within $\mathfrak{T}$	97
$\hat{\mathbf{x}}_{l,k}$	target state estimate where the subscript l indexes the identity within Ξ_k	50
$\mathfrak{T}_k^{\tau_w}$	set of tracks within the temporal window τ_w at k	52
k_s	starting instant of a track within τ_w	52
k_e	ending instants of a track within τ_w	52
$\mathfrak{t}_{l,\mathfrak{K}}^{\tau_w}$	track with identity ξ_l within the interval $\mathfrak{K}_l^{\tau_w} = [k_s, k_e]$	52
$s_l^{\tau_w}$	score assigned to each l^{th} track within τ_w	53
$\mathcal{A}_k$	accuracy error indicating the extent of mismatch between estimated and ground-truth states at k	72
$\mathcal{C}_k$	cardinality error indicating the discrepancy between the number of estimated and ground-truth targets at k	72
METE_k	Multiple Extended-target Tracking Error at k	72
τ_{ltr}	threshold used to calculate the lost-track-ratio	74
$\lambda_d^{\tau_{ltr}}$	lost-track ratio of the d^{th} track at τ_{ltr}	74
$\Upsilon_{\tau_{ltr}}$	number of threshold values used to calculate $\lambda_d^{\tau_{ltr}}$	75
$\text{MELT}_{\tau_{ltr}}$	Multiple Extended-target Lost-Track ratio at τ_{ltr}	75

${}^{g}U_{IDC}$	number of ground-truth tracks with at least one identity change	78
IDC_d^{max}	maximum number of identity changes occurring for the d^{th} ground-truth track	78
NIDC_d	Normalised identity Changes of the d^{th} track	78
C_k	target-intensity map at k	91
$C_{i,k}$	intensity value belonging to C_k	91
$\iota_{b,k}$	energy calculated from the intensity map within $S_{b,k}$	91
r_1	ellipse major semi-axis	91
r_2	ellipse minor semi-axis	91
$\theta_{b,k}$	ellipse orientation at k	91
$\mathcal{D}_k$	set of dummy detections with squared area at k	92
$\mathbf{d}_{i,k}$	single dummy detection within $\mathcal{D}_k$	92
$\tilde{\mathcal{Z}}_k^1$	set of skimmed detections obtained after Non-Maxima Suppression at k	92
τ_{nms}	overlap threshold used for the Non-Maxima Suppression operation	92
$\mathcal{Z}_k^1$	set of detections obtained with the gradient-climbing based detector	93
$\mathbf{z}_{m,k}^1$	single detection within $\mathcal{Z}_k^1$	93
M_k^1	number of detections within $\mathcal{Z}_k^1$	93
$p(\mathbf{z}_{m,k}^1)$	probability density function of $\mathbf{z}_{m,k}^1$	93
$p(\mathcal{Z}_k^1)$	global distribution of $\mathcal{Z}_k^1$	93
$\mathcal{P}(\mathbf{z}_{m,k}^1)$	prior intensity distribution of a single target	93
$\mathbf{z}_{m,k}^{1,h}$	h^{th} proposed detection at h^{th} iteration performed with the Metropolis-Hastings algorithm	93
$\mathcal{H}$	total number of iterations of the Metropolis-Hastings algorithm	93
$F_{m,k}^h$	dynamic model used for the proposal density at k	94
$\nabla \mathrm{C}_k$	2D gradient of C_k	94
$\vec{\mathcal{E}}(\theta)$	normal vectors to the perimeter of the ellipse at θ	94
$\mathcal{Z}_k^2$	set of detections obtained with the detector based on hierarchical-isocontour and morphology at k	95

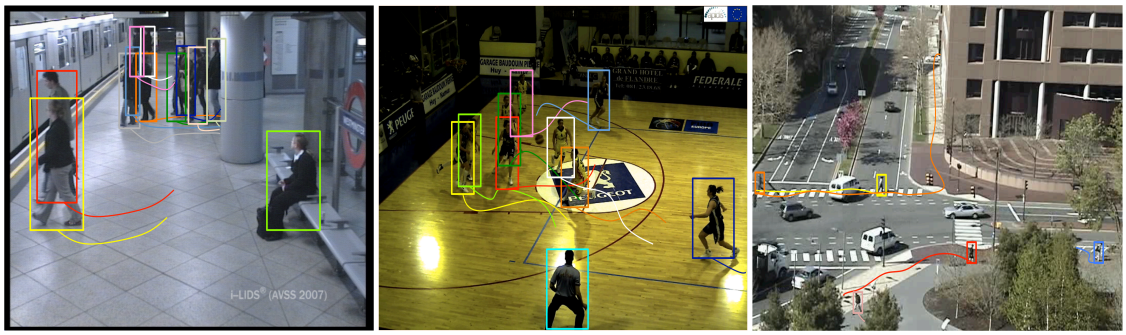
$\mathbf{z}^2_{m,k}$	single detection within $\mathcal{Z}^2_k$	95
M^2_k	number of detections obtained within $\mathcal{Z}^2_k$	95
τ_{iso}	intensity level defining one isoconcontour	95
$\mathcal{I}_{\tau_{iso},k}$	isocontours extracted from C_k at layer τ_{iso} at k	95
$g_{\tau_{iso}}(\cdot)$	function that computes the isocontours at τ_{iso}	95
Ω	set of isocontour levels	96
$G = (E, \mathfrak{T})$	G is a graph: E is the set of edges whose weights are calculated via a link probability and $\mathfrak{T}$ are the nodes	98
$\mathfrak{B}$	buffer size within where the graph matching is performed	99
$\mathfrak{b}$	temporal shift within $\mathfrak{B}$	99
Γ	identity switches per frame	102
Λ	identity switches per track	102
i_k	number of identity switches at k	103
ζ_k	maximum number of identity switches at k	103

Chapter 1

Introduction

1.1 Motivation

The demand for the automated analysis of the behaviour of people, animals and moving objects such as vehicles has grown considerably in recent years. Multi-target video detection and tracking in populated scenes (high density of targets) play an important role in this since they are necessary steps towards fully automated systems [27, 84, 90, 144, 170]. For example, systems for the recognition of human actions and the detection of abnormal behaviours are key to support surveillance tasks [37]. This is achieved by enabling the development of video trackers for motion pattern analysis of single and multiple targets [186]. Single-target trackers help the analysis of motion patterns and behaviours of individuals separately. Multi-target trackers help in quantifying target interactions and comparing motion patterns of different objects simultaneously. Surveillance systems (Fig. 1.1a) use trackers to monitor behaviour [165], to follow selected people and to recognise them in the view of other cameras [27, 117]. The analysis of collective and individual trajectories can be exploited to recognise abnormal behaviours in crowds [146]. Trajectories can be used to recognise interactions among humans [162] and to monitor the activity of people in order to analyse social behaviours [55]; to observe interactions among objects and humans, to help studying collaborative behaviours in meeting rooms [175], or to monitor the position of people with respect to abandoned objects [165]. Tracking is also used in video-based sport analysis (Fig. 1.1b) for automatic summarisation [45] and statistics gathering [71, 167]. In traffic scenes, tracking using fixed or airborne cameras [188] helps the automatic detection of



(a) (b) (c)

Figure 1.1: Examples of video-based applications that benefit from multi-target tracking: (a) surveillance; (b) sport analysis; (c) automatic pedestrian flow monitoring.

unlawful U-turns, vehicles driving in the wrong direction, people crossing roads [72] (Fig. 1.1c) and to collect statistics on typical and atypical behaviours of vehicle and pedestrian flows [16].

In general, the tracking pipeline initially involves a feature extraction stage that generally processes images using prior knowledge of targets (e.g. colour, shape, size) and provides estimated target locations using feature values and classification scores (*confidence maps*) [47, 156]. A confidence map is a (noisy) scalar representation of likely target locations [21, 159, J2] that uses sparse [44] or dense [159] confidence values. Sparse values can be obtained with Support Vector Machines (SVM) through sliding windows [44]. Dense values can be generated with multi-layer homographies [46] or derived from sparse values by low-pass filtering the confidence map [159]. Candidate target locations are then extracted by thresholding and clustering the sparse values with the highest scores [44, 51]. Target locations can also be highlighted by first enhancing the intensity values of targets (*target-intensity map*) and then by localising peaks [138]. Low signal-to-noise-ratio (SNR) maps may lead to erroneous estimations of candidate locations due to unreliable or noisy features. While a confidence map provides likely target locations through feature extraction and classification, in a target-intensity map the target intensities are simply enhanced. Trajectories can then be generated by temporally associating candidate locations with multi-target trackers (tracking-by-detection) [20, 73, 140, 159, 181], directly from confidence maps (track-before-detect) [142,J2] or with generative methods (tracking-by-learning) [182]. Tracking-by-detection approaches address the problem of generating candidate target locations by applying thresholds, clustering and Non-Maxima Suppression (NMS) to the feature values [44]. However, weakly detected features of different parts of a target can generate multimodal confidence values in the target area [51]. Multimodal confidence values

or filter responses can be due to adjacent targets, and hence there is a need to define explicit models to separately detect targets. In the case of high-density scenes with targets having the same or similar appearance, multi-target tracking is addressed with strong priors on target motion [90] and appearance [73]. Alternatively, to deal with low-SNR videos, tracking can be performed directly on confidence maps [J2] or on detections extracted from low-SNR target-intensity maps [140]. Tracking of multiple moving targets can use a point-target representation (e.g. feature-point tracking) or an extended-target representation (e.g. in face or person tracking) [20, 27, 162, 165, 176, 181]. Point-target representations use target position information, whereas extended-target representations also include information about the region covered by the target in the image plane [110,181]. A tracking error can be quantified by computing the discrepancy between estimated and ground-truth target regions [81,124], or employing ground-truth-free tracking evaluation frameworks by enforcing constraints such as time reversibility [149,180] and feature consistency [32,35] of the estimated tracks. In the case of ground-truth based evaluation, the association among estimated and ground-truth tracks needs to be solved [24,28,81,143,187].

In this thesis, we present a method for multi-target tracking based on track-before-detect (MT-TBD) and we evaluate it on real-world surveillance datasets with people. We then present a framework for tracking compact targets, such as bees and ants, where MT-TBD and a novel graph-based data association method for multi-target tracking are evaluated and compared with alternative state-of-the-art approaches that can handle targets in crowded scenarios. Moreover, we present three novel tracking evaluation measures for extended-target models and we compare them with widely used multi-target evaluation measures.

1.2 Problem formulation

The goal of multi-target tracking in videos is to generate accurate estimations of target trajectories (tracks) within the field of view of a camera.

Let $\mathcal{V} = \{v_k\}_{k=1}^{K}$ be a video sequence, where v_k is the k^{th} frame and K is the total number of frames. At time k the feature extraction stage generates a set $\mathcal{Z}_k$ of B filtered features

$$\mathcal{Z}_k = \{\mathbf{z}_{b,k} : k, b \in \mathbb{N}\}, \tag{1.1}$$

where

$$\mathbf{z}_{b,k} = [x_{b,k} \;\; y_{b,k} \;\; S_{b,k} \;\; \iota_{b,k}]^T \tag{1.2}$$

is the b^{th} feature, $x_{b,k}$ and $y_{b,k}$ are the positions with respect to the horizontal and vertical axes, respectively, $S_{b,k}$ is the shape or scale of the feature, $\iota_{b,k} \in \mathbb{R}_{[0,1]}$ is a scalar value between 0 and 1 that indicates the confidence of the feature representing a target, and T is the matrix transpose. Features belonging to the same targets are linked over time in order to estimate tracks. In general, target localisation or association is defined as a function $f(\cdot)$ such that

$$\mathsf{T}_{a,k} = f(\mathcal{Z}_{k-\gamma_1}, ..., \mathcal{Z}_{k+\gamma_2}), \tag{1.3}$$

where $\mathsf{T}_{a,k}$ is the a^{th} track up to frame k and $\mathcal{Z}_{k-\gamma_1}, ..., \mathcal{Z}_{k+\gamma_2}$ are the input features measured in the interval $[k-\gamma_1, k+\gamma_2]$ (measurements), where $\gamma_1, \gamma_2 \in \mathbb{N}_0$. The track $\mathsf{T}_{a,k}$ belongs to the set of tracks $\mathcal{T} = \{\mathsf{T}_a\}_{a=1}^{A}$, where A is the total number of tracks computed within the sequence $\mathcal{V}$. $\mathcal{T}_k$ represents the collection of u_k ($= |\mathcal{T}_k|$ - cardinality of $\mathcal{T}_k$) tracks at frame k. The track of a target a is a time series

$$\mathsf{T}_a = \{\hat{\mathbf{x}}_{a,k} : 1 \leq k \leq K\}, \tag{1.4}$$

where $\hat{\mathbf{x}}_{a,k} \in \mathbb{R}^n$ is the state of the target at frame k and n represents the dimension of the state. The information encoded in the state $\hat{\mathbf{x}}_{a,k}$ is used to describe the status of the target at k. The definition of $\hat{\mathbf{x}}_{a,k}$ is application-dependent, in fact $\hat{\mathbf{x}}_{a,k}$ may encode the position and velocity of the target [142], or also shape information, such as width and height of the target [43]. For example, a target on the image plane can be represented by its 2D-position with a certain shape information and an identity ($n = 4$),

$$\hat{\mathbf{x}}_{a,k} = [x_{a,k} \;\; y_{a,k} \;\; S_{a,k} \;\; \xi_a]^T, \tag{1.5}$$

where $x_{a,k}$ and $y_{a,k}$ represent the target position on the horizontal and the vertical axes, respectively, $S_{a,k}$ is the target region information on the image plane and ξ_a is the target identity (ID). $S_{a,k}$ may be represented in the form of a bounding box [27], a bounding ellipse [181] or a bounding contour [164]. In the case of point targets, the estimated state of the target a does not contain $S_{a,k}$ and it is denoted as $\hat{\mathbf{x}}'_{a,k}$. Associated to $\mathcal{T}$, $\hat{\mathbf{x}}_{a,k}$, $\hat{\mathbf{x}}'_{a,k}$, u_k, ξ_a, T_a we have the corresponding ground-truth track information that is defined as ${}^g\mathcal{T}$, ${}^g\hat{\mathbf{x}}_{d,k}$, ${}^g\hat{\mathbf{x}}'_{d,k}$, ${}^g u_k$, ${}^g\xi_d$, ${}^g\mathsf{T}_d$. The ground-truth information is used to evaluate the accuracy of the tracking estimation [J1].

(a) (b)

Figure 1.2: Example of colour variations of a target due to illumination differences: (a) in a shop; (b) in a corridor.

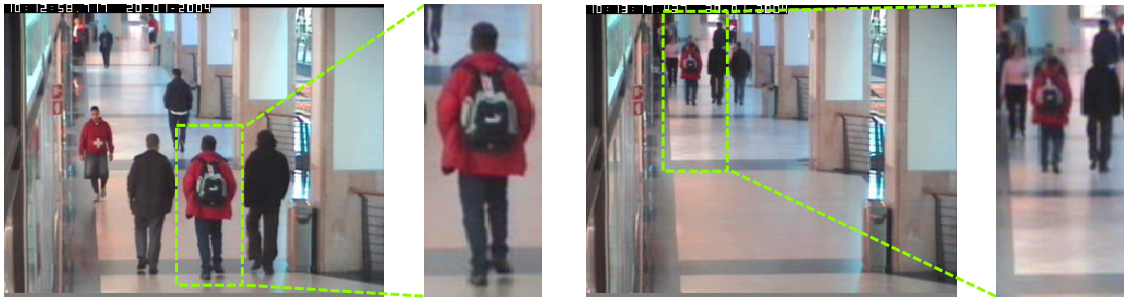
Figure 1.3: Example of clutter: the person to be detected is not clearly distinguishable due to appearance similarity with the background and with other objects (e.g. a mannequin).

1.3 Challenges

The challenges a tracker may face involve colour similarities among objects, illumination changes, pose variations, size changes, occlusions, various noise components, abrupt or unpredicted motion variations, and the density of targets in the scene.

Colour similarities can mislead the target-background and the target-target discrimination. When another region in the image has similar colour to that of a target, then a track can be lost. Similarly, when targets with similar colour move close to each other, their identities can be swapped [95,181]. *Illumination* changes caused, for example, by different light sources (Fig. 1.2) lead to colour variations that can induce target losses. This problem can be addressed by using illumination invariant features or by updating the colour model of the targets [148].

Shape similarities can also generate ambiguities (Fig. 1.3) between a target and an object in the background, or among similar targets [179]. Examples include people tracking when the shape is encoded as the head-and-shoulder or full-body outline [51]. *Pose* variations leading to shape changes (Fig. 1.5) require a tracker to be capable of adapting the corresponding appearance models to avoid track inaccuracies or losses [111].



(a) (b)

Figure 1.4: Example of size variations of a target due to the camera perspective distortion: (a) target area at close view; (b) same target area at far view.

(a) (b)

Figure 1.5: Example of shape change due to pose variation: (a) front view; (b) side-rear view.

Figure 1.6: Example of a largely occluded person (man with the black jumper).

Size changes which are due to either perspective distortion (Fig. 1.4) or camera zoom can lead to an erroneous target localisation. Similarly to the case of pose variations, tracking also has to adapt the shape model (e.g. bounding box size) in order to extract correct appearance features from the whole target area.

When *occlusions* happen (Fig. 1.6), the only data available to the tracker are the measured target dynamics and the appearance features before (and after) the occlusion itself [27, 181]. Using this information, a tracker can estimate the likely location of a target by interpolating spatio-temporal data when the target is not observable.

Noise components can be introduced during video acquisition or compression, and may lead to corrupted measurements that generate unreliable features for the estimation of the target locations.

Motion-related challenges can be due to abrupt variations (e.g. sudden accelerations) or unusual dynamics (e.g. deviations from a predicted path to avoid obstacles). Most tracking algorithms rely on prior models for motion prediction [27]. These models are mainly linear with additional terms that represent small variations as noise components [142].

Finally, the difficulty of tracking depends on the *density* of targets in the scene. In the case of people tracking, when the crowd motion tends to be coherent in one direction and does not vary over time because of the high spatial density of people, crowded scenes are defined as *structured*. In *unstructured* crowded scenes, instead, groups of people may move simultaneously in different directions [144]. Successful trackers rely on part-based detectors and perform non-causal tracking using target re-identification [95].

1.4 Contributions

Given real-world video sequences containing multiple targets, our aim is to estimate their locations over time (trajectories) and to evaluate the accuracy of the estimated trajectories. The proposed methods are evaluated on surveillance and biology scenarios. Both scenarios assume extended targets being independent of each other. In the former we assume that the detections are provided, whereas in the latter detections are not provided and targets have indistinguishable appearance. We do not assume any specific motion property of the targets other than that the targets follow a linear motion within a short interval of time (e.g. $\sim$3 frames). The estimated trajectories are evaluated with respect to ground-truth information by using the set of new evaluation measures.

The main contributions of the thesis are the following:

1. Online multi-target *track-before-detect* (MT-TBD) applied on confidence maps used as observations and assumed to be provided. The proposed tracker is based on particle filtering and automatically initialises tracks. The main novelty is the inclusion of the target ID into the particle state, enabling the algorithm to deal with unknown and large numbers of targets. The problem of mixing IDs of targets close to each other is addressed using a probabilistic model of target birth and death based on a Markov Random Field applied to the particle IDs.

Each particle ID is managed using the information carried by neighbouring particles. The assignment of the IDs to the targets is performed using Mean-Shift clustering and supported by a Gaussian Mixture Model [J2]. The output of MT-TBD is postprocessed by introducing latency that allows the algorithm to link short trajectories and to discard spurious ones. We qualitatively show the effectiveness of embedding the ID inside the particle state to deal with partially occluded targets and limitations when targets undergo full occlusions. The proposed solution outperforms an alternative state-of-the-art solution throughout a range of increasing latencies of postprocessing.

2. Three parameter-independent measures for evaluating multi-target video tracking which assume that the ground-truth data are provided[1]. The measures take into account target-size variations, combine accuracy and cardinality errors, quantify long-term tracking accuracy at different accuracy levels in terms of lost-track-ratio, and evaluate ID changes relative to the duration of the track in which they occur. We show that these measures enable a more detailed assessment of the tracking performance than alternative state-of-the-art measures and can objectively evaluate tracking performance without relying on preset parameters.

3. Multi-target detection and tracking applied on low signal-to-noise-ratio images containing a high-density targets[2]. We propose a gradient-climbing technique and an isocontour slicing approach for intensity maps to localise targets with indistinguishable appearance. The former uses Markov Chain Monte Carlo to iteratively fit a shape model onto the target locations, whereas the latter uses the intensity values at different levels to find consistent object shapes. Trajectories are generated by recursively associating detections with a greedy graph-based tracker on time-shifting windows. The edges of the graph are weighted with a likelihood function based on location information. The proposed localisation methods outperform alternative solutions both in detection and tracking. As far as tracking is concerned, the major improvement is achieved in terms of ID maintenance on targets.

1.5 Organisation of the thesis

The report is organised as follows:

Chapter 1: Introduction to multi-target tracking and its application to real-world problems. For-

[1] This work appears in [J1]. More details about my contribution are provided in Chapter 4.

[2] This work was submitted to IEEE Transactions on Circuits and Systems for Video Technologies and it is currently under review.

mulation of the tracking problem along with the challenges that can be encountered on real-world scenarios. List of contributions of the work presented in the thesis.

Chapter 2: Definitions of different levels of crowd densities, scale of observation and overview of feature extraction methods in videos. Presentation of multi-target tracking methods by analysing their main stages of processing. Definition of causal and non-causal methods. Discussion of the predictive models used to estimate the location of targets. Description of sequential estimation and batch association to generate tracks. Analysis of different methods for track initialisation and track termination, and description of the use of contextual information to facilitate multi-target tracking. Evaluation measures to quantify the performance of multi-target trackers in the case of point and extended targets. Finally, discussion of the limitations of multi-target trackers and performance evaluation measures.

Chapter 3: Method for multi-target tracking on confidence maps using Markov Random Fields into the Bayesian formulation to keep targets separated during tracking. Experimental validations and sensitivity analysis of the method pipeline.

Chapter 4: Evaluation measures to calculate tracking performance in terms of frame-based accuracy, long-term tracking and robustness to ID switches. Comparison of trackers using the presented measures and comparison of these measures with state-of-the-art measures.

Chapter 5: Feature extraction method to detect compact targets on low signal-to-noise-ratio images and a graph-based data association algorithm to perform multi-target tracking. Results about the sensitivity of the methods and comparison with alternative approaches from the state of the art.

Chapter 6: Summary of the achievements and future research directions.

Chapter 2

State of the art

2.1 Introduction

Tracking of multiple moving targets may involve point-target representations (e.g. feature-point tracking) or extended-target representations (e.g. in face or person tracking) [20, 27, 162, 165, 176, 181]. Point-target representations use target position information, whereas extended-target representations also include information about the region covered by the target in the image plane [110, 181]. The tracking pipeline can be divided into four main stages (Fig. 2.1): detection (composed of region detectors and feature extraction), localisation or association (which exploits features to identify the position of the targets on the image plane), prediction (which models the motion of targets to predict their future locations), and track postprocessing (to refine tracking results). Moreover, the output tracks can be used to extract contextual information by learning the environment [109] and by updating the model of the targets in order to boost the localisation of targets [90, 148].

We perform an analysis of the state-of-the-art that is mainly focused on algorithms employing extended-target representation. Firstly, for each stage of the pipeline, we aim to describe alternative solutions in order to infer effective tracking properties for dealing with challenging problems, such as identity maintenance on targets in the case of high-density of targets. Then, we analyse multi-target tracking evaluation methods in order to show that the literature is still lacking a framework to enable a holistic analysis of the tracking performance independent of applications and in a parameter-free manner.

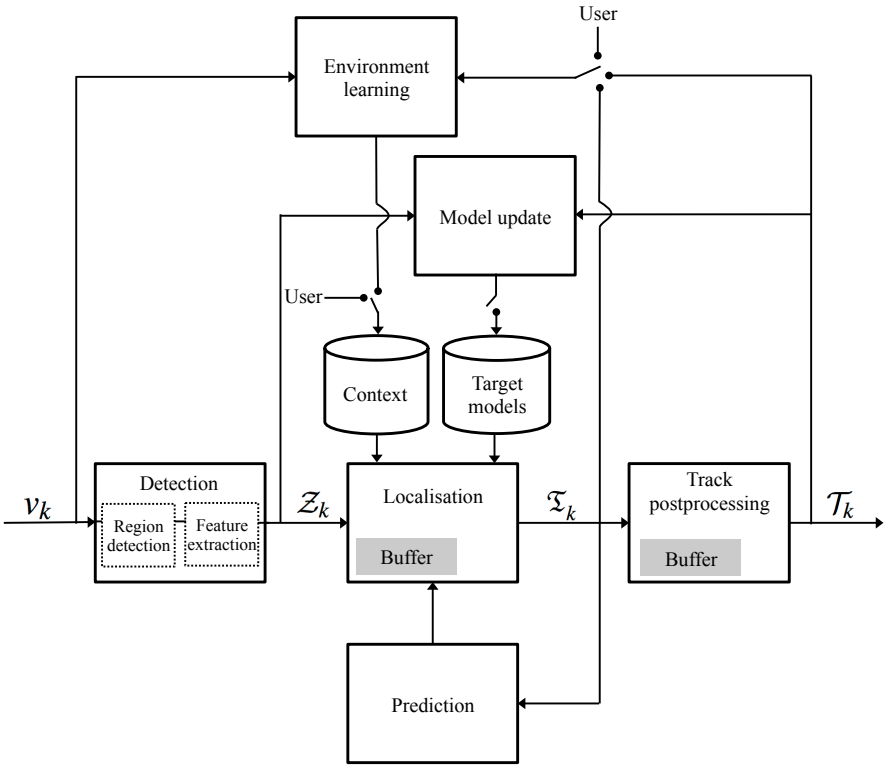


Figure 2.1: Block diagram of a tracker with sequential localisation. The buffer accumulates measurements to allow processing over temporal windows. The localisation stage can be externally initialised or can use contextual information, such as a map of the scene, and target model. The track postprocessing stage improves the quality of the final result, for example by linking short tracks or by deleting spurious tracks.

In this chapter, we present the state of the art for feature extraction that addresses extended-target detection, multi-target tracking methods and procedures to assess their performance in videos containing extended targets. In Sec. 2.2 we characterise scenes based on the crowd density and define the scale of observation. In Sec. 2.3, we briefly overview methods for detection by dividing them into methods for region detection and feature extraction. The literature for multi-target tracking methods (Sec. 2.4) analyses the parts related to prediction, localisation and association, initialisation and termination, and scene contextual information. In Sec. 2.5, we review methods for the performance evaluation of multi-target tracking methods in the case of extended targets. Finally, Sec. 2.6 draws conclusions about the described methods.

2.2 Definitions

In this section, we provide definitions of crowd-density while giving examples of applications in different contexts. Following that, we define the levels of observation that constrain the video analysis on scenarios having different densities and view points.

(a) (b) (c)

Figure 2.2: Representative samples of crowd densities: a) *high-density* crowd, b) *mid-density* crowd, c) *low-density* crowd.

2.2.1 Crowd-density models

The crowd density can be divided into three main categories: low-density (i.e. a scene with few clearly distinguishable people), mid-density (i.e. enough people where occlusions are frequent but there is room to move) and high-density (i.e. large number of people who are close to each other and indistinguishable) crowds (Fig. 2.2).

In *low-density* scenes people are modelled as single entities [129, 150, 165, 185] and the detection stage is a very important step because it must provide reliable candidates to the tracking stage. Sets of state vectors describing the moving people in the scene (e.g. position and velocity) are then generated via tracking [129, 150]. For example, change detection algorithms for the detection of moving people [137] combined with Bayesian tracking [J2] are used in these situations. The analysis of human interactions, such as meeting, walking together and splitting, can be performed using information retrieved from the estimated state vectors [129, 162, 165], and consistent target identities must be computed at the tracking stage to ensure discrimination of each individual over time.

People in *mid-density* crowds are modelled with two different approaches: local recognition of single entities [26,184] and global motion analysis [13,118,133,151]. In the former, detections need to discriminate people well since occlusions are highly likely to occur [102, 184]. For example, multiple target cues driven by detection can be combined and kept updated over time in order to enhance the discriminative capability of a model of a person with respect to adjacent people [184]. The person-model adaptation (model update) for detection enables supervision to the single object trackers. Since the appearance of the person can gradually change over time, the amount of lost tracks can be minimised thanks to the update stage. In order to avoid drifting due

to the model update, in each target cue authors included the domain knowledge (e.g. the features' position in the space) and the update was performed only with reliable detections. As far as methods based on motion analysis are concerned, the crowd motion can be modelled with vector fields in terms of principal direction, speed and crowd mobility [151]. For example, Andrade *et al.* [13] considered the crowd as a global entity and the whole motion of the scene was used to find pattern similarities over time. Mehran *et al.* [118] proposed a method based on social forces [63] to model human interactions within a crowd. The scene was described by a dense force flow that was generated by the moving people. Recently, approaches have merged the concepts of scene motion analysis and recognition of single entities [85, 102]. The vector field approximating the global motion can be extracted from human trajectories using spline interpolation [85]. Similarly, motion patterns can be spatially segmented from trajectories using driving force models in order to enable the recognition of human groups having similar behaviours [102].

High-density crowds can be modelled as fluids and analysed using fluid dynamics equations [11, 76, 191]. A dense crowd can be modelled as clouds of particles transported under the action of the flow field generated by the crowd motion [11]. A fluid that, for instance, flows in a pipe is subjected to constraints such as barriers. The same concept can be extended to a moving crowd that has to deal with a physical barrier in the scene. This consideration leads to the understanding of the scene in terms of (i) heading direction, (ii) number of crowd segments and (iii) locations at which segments merge or split. The flow field generated from the motion of a crowd can also be described in terms of divergence (representing dispersion) and vorticity (linked to moving around obstacles) [41]. When motion is described through these two terms, the detection of obstacles becomes straightforward. Hughes *et al.* [76] formulated three hypothesis in addition to modelling a crowd as a fluid: i) the speed of a person depends upon the surrounding people, ii) people can have common destinations and iii) people seek to minimise their estimated travel time. With these hypothesis, Hughes *et al.* [76] defined the basic governing equations for the flow of a single person type. In this way it is possible to have a crowd of various types of people walking toward different objectives or with various speed relationships. Assuming that people behave according to optimal strategies (i.e. in a standard situation they do not make complicated decisions between various alternative behaviours, but simply react to obstacles or other people) when they are in a large crowd, it is possible to model their behaviour using so called *social forces* [63]. The forces that affect people are caused by the repulsive inter-reactions that are

generated among people within the crowd. Although the motion of the people is affected by the environment (i.e. other people or barriers), the aims, destinations and dynamics in a crowded scene are fairly predictable. Wang *et al.* [176] modelled the crowd locally, characterising each pixel by its location and direction of motion. The information retrieved from the motion of the objects (i.e. pedestrians and cars) was used to classify common patterns. Authors in [70, 89] proposed a localised model of the scene. Kratz and Nishino [89] divided the scene into blocks and for each block considered volumes of motion over time. Hence, the description of the crowd motion was contained and localised within fixed spatio-temporal volumes. Hu *et al.* [70] did not consider volumes but modelled the whole crowd in terms of motion direction.

2.2.2 Scale of observation

According to [118], models for crowd analysis can be microscopic, mesoscopic and macroscopic. In *microscopic* scenarios it is possible to reach high resolution levels to distinguish single individuals and hence perform interaction analysis relying on their behaviours. The analysis is performed considering humans as independent moving entities and their interactions are treated as a self-organisation processes [62]. With the collection of individual behaviours, recognising interactions is fairly straightforward [162, 165]. In *macroscopic* scenarios, it is exceptionally hard (almost impossible) to reach high reliability and be able to distinguish individuals. The individual body details (on occasions when they are fully visible) are not high enough to distinguish and associate them to a single person. It was demonstrated that it is better to model such scenarios as problems of fluid dynamics and treat humans as particles in a fluid [11, 41, 76]. The analysis of interactions is kept at a high level, and it mainly focuses on goal-oriented behaviours. *Mesoscopic* scenarios [118, 193] inherit both previous models and the methods aim to study interactions among subgroups of people. It is still not clear which models perform best for the retrieval of motion patterns. In fact, classes of methods in the literature are twofold: (i) based on multi-person tracking [20, 27, 93, J2] and (ii) based on particle representation of people [118].

There is no sharp distinction of where a scenario needs to be treated with mesoscopic or macroscopic models. The principal elements to take into account are the physical boundaries imposed by the resolution of the cameras. In particular, there are some constraints that force the employment of specific models for the analysis of scenes. Johnson's Criteria [88] defines four levels of discrimination for pixel resolution (Tab. 2.1). The reported numbers come out of a military context, but are also applicable to general cases, and specifically applied to the retrieval

Table 2.1: Four levels of discrimination for the specific tasks in head information retrieval.

Task	Area (pixels×pixels)	Description
Detection	$\approx 2 \times 3$	Head presence
Orientation	$\approx 3 \times 4$	Symmetrical, horizontal or vertical
Recognition	$\approx 8 \times 10$	Object typology distinguishable, e.g. person vs. car
Identification	$\approx 13 \times 15$	More detailed object description, e.g. man vs. woman

of heads in images. They are considered highly optimistic and estimated in *quasi*-ideal cases, like a featureless background with ideal illumination conditions. In the presence of clutter, when signal to noise ratio increases, such numbers might consequently increase.

2.3 Detection

The detection process is divided into region detection and feature extraction for the generation of object descriptors. Detected regions can be elliptical and invariant to affine transformations [119]. For example, Maximally Stable Extremal Regions (MSER) can be used to define regions having nearly the same support [115]. MSER are connected components generated from thresholded images and can be used to detect objects in a scene [119]. Regions can also be fixed (e.g. rectangular) and dense features, for example Histogram of Oriented Gradients (HOG) [44], can be extracted from the them [44].

Descriptors can be generated using *colour* histograms [94, 95, 135, 181, 189]. Colour histograms are used to distinguish targets over time while addressing the challenges of targets with similar colour. In order to reduce sensitivity to light variations, histograms are generally quantised with 8 bins per Red-Green-Blue (RGB) channel [94]. *Shapes* can be used to represent targets, for example in the form of HOG. This representation is popular for describing heads [20] and bodies [27, 159]. Alternatively, *edgelets*, a large pool of short lines and curve segments (based on intensity gradients), can be used to represent human shapes [179]. This method employs descriptors for head, torso, leg and full body that are combined to address the problem of occlusions. Similarly, *shapelets* are combinations of oriented gradient responses learned in a discriminative manner on local patches [147]. Targets can also be described with covariance matrices as *texture* descriptors. In this case, a dense model of covariance features (e.g. spatial location, intensity, higher-order derivatives) is used inside a detection area [169]. A target can be represented with several covariance descriptors of overlapping regions, where the best descriptors are determined with a greedy feature-selection algorithm combined with boosting.

The covariance matrix descriptor is applied on image patches to characterise and distinguish targets [94, 95]. The Scale-Invariant Feature Transform (SIFT) [108] can also be used to capture texture characteristics in order to describe, for example, human torso regions [184]. Textures can also be described through non-parametric grey-scale invariant primitive statistics called Local Binary Patterns (LBP) [60, 61]. LBP has the advantage of tolerating considerable grey-scale variations (e.g. illumination) so that no normalisation of input images is needed. Alternatively, Binary Robust Independent Elementary Features (BRIEF) can be used to describe objects with simple pair-wise pixel differences while allowing an inexpensive computation [31].

When the scene background is fixed, it is possible to detect moving targets by calculating the difference between the current frame and a reference background frame (*background subtraction*) [131]. One can use a simple weighted frame difference between the actual and the background frame [18, 37], or a mixture of Gaussians [190] where each component of the mixture belongs to a colour channel (e.g. three components for RGB). In this case, the mixture of Gaussians is learned on the background and the probability of each pixel of a new incoming frame being considered a part of background or of target is then calculated. *Optical flow* [163] can also be used to extract candidate target location of moving targets [33, 98, 106]. The goal of this technique is to find a motion field that describes the target motion in the scene and to discard static objects. Targets can then be accurately detected within regions with motion by using appearance features, such as colour or shape [33, 98, 106].

2.4 Multi-target tracking

Tracks can be estimated either by extracting features in each frame and linking them over time, or by extracting features at initialisation (i.e. when a target appears in the scene) and then by letting the tracker generate future location hypotheses (prediction) and confirm them over time (update) using the newly extracted features. The choice of the type of features provided to the localisation or association stages is important regarding the use of a tracker with static or moving cameras [47, 97]. If the feature extraction relies only on target information, such as outline of targets [179], the tracker can be extended to moving camera applications [95]. However, if the feature extraction relies on the background information (e.g. using background subtraction [190]) and if the localisation stage is highly dependent on this feature, major modifications to the localisation stage are needed to extend the tracker from static to moving cameras. An adaptation

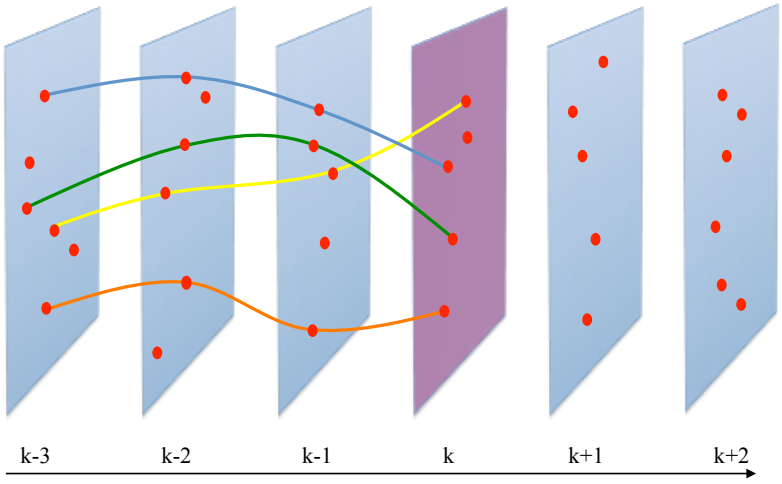
to moving cameras is also needed when contextual information (e.g. entry/exit points [181]) is included in the localisation stage.

In the following sections we provide an overview of multi-target tracking methods by classifying them into causal and non-causal trackers. We then describe the predictive models used by the trackers to generate target location hypotheses. This description is followed by an analysis of localisation and association methods that enable the estimation of the position of targets. Finally, we discuss how the initialisation and termination of trackers is performed, and how the contextual information is employed to aid tracking.

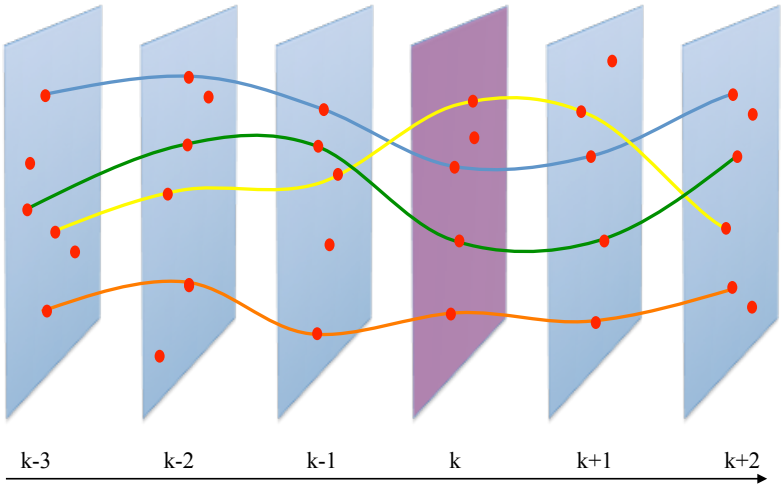
2.4.1 Overview

Some approaches formulate the problem of simultaneously tracking a number, A, of targets as a problem of single-target tracking, A times. The target-tracker association is performed by an external algorithm that guarantees that one tracker is exclusively associated with a target [65]. Alternatively, multi-target tracking can be formulated as the problem of jointly tracking all the targets by using a single tracker [43]. When the number of targets increases, maintaining the identities of all the tracks correctly associated with the targets becomes challenging. Interactions among neighbouring targets can be modelled in order to maintain the identities of the right targets [65, 83].

Trackers can be causal or non-causal filters (Eq. 1.3). *Causal trackers* $(\gamma_1 > 0, \gamma_2 = 0)$ (see Sec. 1.2) only use features extracted from the past and the current frame k to estimate tracks (see Fig. 2.3). Causal trackers, such as tracking methods based on particle filtering [17] and Markov Chain Monte Carlo (MCMC) [14], are used for time-critical applications. *Non-causal trackers* $(\gamma_1 \geq 0, \gamma_2 > 0)$ use also future frames, thus resulting in a delayed decision (Fig. 2.3). Non-causal tracking [19, 74, 188, 189] is typically formulated as a global optimisation problem to retrieve target tracks throughout the video sequence [139]: the candidate target locations for the whole sequence [36] are obtained at the feature extraction stage and are then linked together using optimisation processes [74, 101]. Motion models are implicitly included into the optimisation algorithm and they are commonly expressed as constant velocity models [36]. Non-causal methods can be divided into two categories: (i) methods that iteratively compute long tracks by associating time-independent features and (ii) methods that build long tracks in multiple steps, by extracting short-term tracks either with causal or sub-optimal association trackers, and then by associating shorter tracks (i.e. tracklets) into longer tracks. Examples of non-causal trackers [135]



(a)



(b)

Figure 2.3: Causal and non-causal multi-target tracking: (a) causal trackers operate using measurements from the current and past instants; (b) non-causal trackers generate the results using past, current and future observations.

include detection association trackers such as Multiple Hypothesis Tracking (MHT) [139] and HybridBoosted tracker [103].

2.4.2 Prediction

Predictive models generate target hypotheses that the localisation stage validates using current measurements [110, 142]. Predictive models can use, for example, kinematic equations (e.g. constant velocity) [142] or motion estimation models (e.g. [121]), and can involve a training phase of the target evolution [90]. Learning-based models mostly exploit a time interval at the beginning of a video sequence for training [12]. One can calculate the state $\hat{\mathbf{x}}_{a,k}$ based on the predicted state $\tilde{\mathbf{x}}_{a,k}$ (see Sec. 1.2) and the current measurements [142]. The predicted state $\tilde{\mathbf{x}}_{a,k}$ is

calculated using a function $F_{\mathbf{x}}(\cdot)$ applied on the state at the previous frame $k-1$, such that

$$\tilde{\mathbf{x}}_{a,k} = F_{\mathbf{x}}(\hat{\mathbf{x}}_{a,k-1}). \tag{2.1}$$

The function $F_{\mathbf{x}}(\cdot)$ is also known as the motion model or evolution model. The motion model is used to draw state hypotheses from the current frame to the next, mostly using kinematic models [142]. These hypotheses are further validated using features extracted from the current image frame. Hence, the state $\hat{\mathbf{x}}_{a,k}$ is estimated using the previous state $\hat{\mathbf{x}}_{a,k-1}$ and the measurements from the image at frame k. Motion models can be either pre-learned [12, 65, 90, 101, 144] or fixed [10, 18, 20, 27, 43, 68, 71, 128, 159, 173, 184, 190].

Autoregressive motion models are used by particle filter algorithms [17] for linearly predicting future target locations [10, 27, 43, 65, 68, 128, 159]. Equation 2.1 for a generic autoregressive motion model takes the following form:

$$\tilde{\mathbf{x}}_{a,k} = F_{\mathbf{x}}\hat{\mathbf{x}}_{a,k-1} + \omega_{k-1}, \tag{2.2}$$

where $F_{\mathbf{x}}$ is an $n \times n$ matrix defining the linear function $F_{\mathbf{x}}(\cdot)$ and ω_{k-1} is random noise with a given distribution (e.g. Gaussian). Motion estimation algorithms often generate the motion flow between consecutive frames [121], which in turn can be exploited to build predictive models. A predictive motion model for consecutive features can be designed using a constant velocity model with the contribution of Kanade-Lucas-Tomasi (KLT) point tracks [20, 145, 168]. In particular, the prediction state is defined as

$$\tilde{\mathbf{x}}_{a,k} = \hat{\mathbf{x}}_{a,k-1} + \eta\ \tilde{\mathbf{v}}_{a,k-1}, \tag{2.3}$$

where $\tilde{\mathbf{v}}_{a,k-1}$ is the velocity estimation coming from the KLT tracks in the frame prior to the current state and η is the time interval between the states where the velocity is calculated.

Mode-seeking trackers such as the Mean-Shift tracker [38] follow neighbouring modes of clusters generated with features extracted from the frames. Clusters are represented as modes and tracking is performed by seeking the closest mode in the subsequent frame [18]. The predicted location of the target in the subsequent frame lies in the area defined by a kernel that is dependent on the target position in the previous frame. Each mode displacement is therefore assumed to be smaller than the kernel size.

Learned models are used to improve performance when motion is predictable, for example, in the case of high target density [12,144]. Assuming that each target follows a coherent direction with respect to the other targets, it is possible to learn motion models and to include them into the tracker to help the prediction of target positions. For example, in crowded scenes a set of motion constraints can be trained from the behaviour of humans [12]. These motion constraints are properties retrieved from entry/exit regions and common paths of people, influences generated by barriers or walls, and the behaviour of people surrounding the tracked person. The tracker can rely on a grid of particles over the image plane and tracking can be performed by maximising the transition probability of a particle from one cell to another. The transition probability can be determined by two factors: (i) the colour similarity between the current and the next location and (ii) the influence of the learned motion constraints in this location.

Scene dynamics can be learned using optical flow features [121] (i.e. position and velocity) and encoded according to a codebook, where each word of the vocabulary is associated to a specific dynamic [144]. The target location is computed using a weighted mean of the displacement of the observations based on the learned dynamics and the predicted displacement. Alternatively, time-varying dynamics of people across different spatial locations can be modelled using Hidden Markov Models (HMM) [90, 136]. The hidden states of the HMM encode possible motion patterns that are likely to be present at each spatial location.

2.4.3 Localisation and association

Localisation and association stages rely on the measurements generated by the feature extraction stage and validate feature similarities over time to estimate reliable tracks. The validation can be performed sequentially or as a batch process (Fig. 2.4). Sequential localisation extracts tracks recursively, whereas batch processes optimise links among features collected within a time interval to generate tracks.

Sequential localisation

A Particle filter recursively finds targets using Bayesian recursion for the sequential estimation of the target states over time [78, 142]. The Bayesian recursion involves the estimation of the target state[1] $\hat{\mathbf{x}}_k$ calculated by constructing the posterior probability density function (pdf) using motion models (prior distribution) (see Sec. 1.2) and measurements (likelihood function) gathered

[1]The subscript a has been removed to generalise the problem.

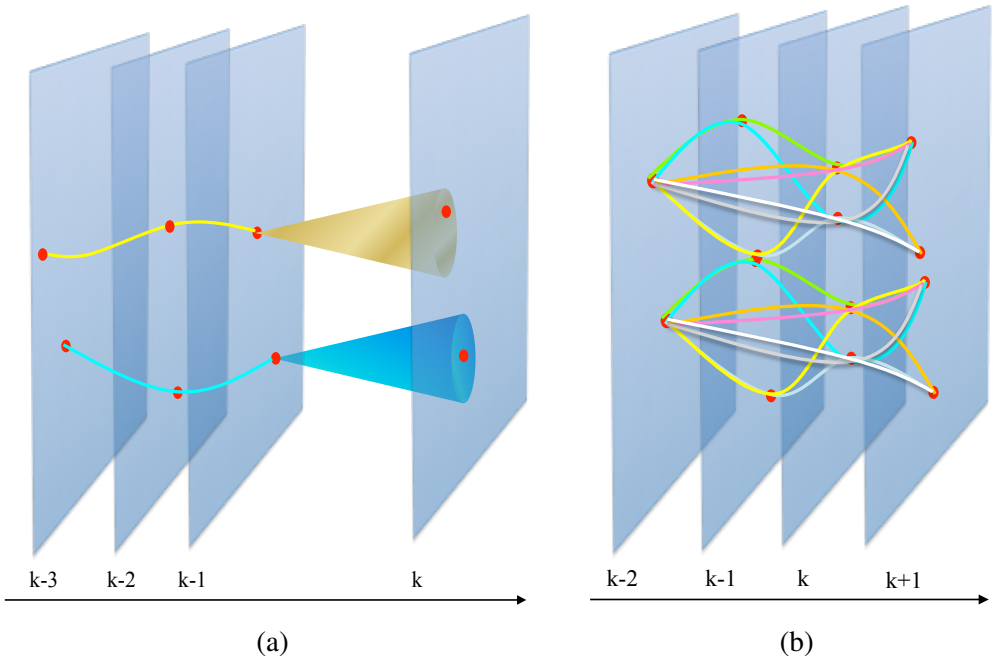


Figure 2.4: Comparison between a sequential localisation and a batch association approach. (a) Sequential localisation uses predictive motion models to explore potential target areas. (b) Batch association generates track hypotheses (coloured lines) that are selected based on an optimisation procedure.

from the current frame. The prior is often calculated by neglecting the most recent measurements as in the CONDENSATION algorithm [34, 78]. The posterior pdf can be a multimodal distribution, where the modes of the distribution represent likely target locations. In order to make the Bayesian recursion computationally tractable, the posterior pdf is approximated with a Monte Carlo method [48], which consists of a set of random samples, or particles, drawn from the posterior pdf with associated weights.

The extension from single to multi-target particle filter requires the size of the state to be made proportional to the number of targets, i.e. $\hat{\mathbf{x}}_k \in \mathbb{R}^{n'}$ where $n' = n \cdot u_k$, with u_k being the number of targets at k [75]. Generally, when a single particle filter has to deal with multiple targets and the distribution of the states is represented with a mixture of distributions, one of the major problems is the maintenance of the multi-modality [171]. Hence, a mechanism based on AdaBoost can be included into the tracker in order to maintain the multi-modality [128]. Alternatively, a baseline version of the particle filter is applied on confidence maps generated with Cascade Confidence Filtering which incorporates constraints on the size of the objects, on the background importance and on the smoothness of trajectories [159]. A feature extraction stage relying on geometric structures along with background filtering produces preliminary confidence

maps for a certain time interval. Spurious features within this time interval are filtered out with a temporal smoothing method based on the Vessel filter [52]. The resulting confidence maps are used as observations for the multi-modal particle filter. Lastly, trackers can also rely on the detections generated, for example, by an AdaBoost classifier [128, 172].

Trackers can be composed of multiple particle filters, each operating on one target [10]. The communication among particle filters can be carried out with a heuristic method relying on the spatial locations of the extracted features. Each detection is associated to the spatially closest trajectory, whereas each unassociated detection leads to the generation of a new trajectory. When particle filters are considered dependent, the association among detections and trajectories can also use past state estimates of multiple particle filters as observations in order to aid tracking [65]. Features are combined with the probability of a target being in a certain location with respect to (i) its predicted state estimate, (ii) colour similarity with respect to trained templates, (iii) dissimilarity with the background, (iv) penalty scores with target regions overlapping each other and (v) neighbouring targets. Furthermore, the Hungarian algorithm [92] can be used to associate particle filters to detections [184]. Here, the assignment matrix is constructed using a Bayesian formulation among features and track states. The association between features and tracks can alternatively be performed with greedy algorithms [27]. A single particle filter is employed for each target and an on-line AdaBoost classifier is trained for each target against all the others in order to better discriminate the tracked targets.

Random Finite Sets (RFS) can also be used along with the Bayesian formulation in order to perform multi-target tracking [110, 112]. RFS treat states and measurements as realisations of random variables and the Bayesian formulation with RFS can be approximated with Monte Carlo methods [68]. The feature extraction stage can be embedded into the tracker to define the appearance model and, like the motion model, it is defined a priori.

Markov Chain Monte Carlo (MCMC) methods can alternatively be used to draw samples from posterior distributions since, with a particle filter, it is very hard to deal with large number of targets and hence large state spaces. In fact, maintaining the multi-modality requires very precise mathematical methods [171] and the computational cost for handling high-dimensional state spaces is still prohibitive. For these reasons, in order to avoid expensive integration steps, MCMC methods have been introduced [14]. For example, Smith *et al.* [157] used MCMC to deal with 10-dimensional states. Moving humans can be represented with 3D models using camera

calibration parameters after being detected via background subtraction. The information about their locations is employed to build a multi-person joint likelihood function and used to find person locations in consecutive frames, leading to a dimensionality of the space proportional to the number of persons in the scene [190]. Kalman filters are then used to build the posterior pdf for consecutive frames employing a fixed motion model describing people moving with constant velocity and affected by Gaussian noise. Since a joint likelihood is used, which involves both discrete and continuous variables, MCMC is employed to sample from the posterior pdf and to obtain the estimation of the target states calculating the Maximum A Posteriori (MAP). Alternatively, track hypotheses can be extracted within a four-second window using Minimum Description Length [20]. Features such as scale, location and motion computed with Kanade-Lucas-Tomasi (KLT) are associated over time using likelihood functions. A refinement stage relying on the likelihood functions is built to allow two types of modifications to track pairs, namely the move of certain features from one track to the other or the swapping of all the features belonging to both tracks at a chosen time instant. MCMC is then used to make decisions about the acceptance of such modifications and to confirm the final track decision.

Finally, sequential localisation can be performed with ad-hoc methods, either based on thresholds or on combinations of different algorithms. For example, the link between two features can be defined by a probability (i.e. the link probability) calculated as a product of three independent affinities calculated from feature characteristics, such as position, size and appearance [74]. The final linking between two features is then confirmed by using a two-threshold strategy. The first threshold is used to check if the link probability is high enough; whereas the second threshold is used to determine if the affinity of any of their conflicting pairs is high enough. Alternatively, a template-matching approach can be used in dense crowds leveraging the prominence of clearly distinguishable people (determined by the extracted features) and using social forces [63] to bound the movement of people due to limited room within the crowd [77]. The tracking framework updates first the positions of prominent people and then the positions of people with lower prominent features. The incorporation of the influence of neighbours is crucial to achieve a reliable tracking in scenes with high-density crowds in order to avoid tracking drifts.

Batch association

Features can be associated over time with a batch process through maximisation algorithms applied on posterior probability, which quantifies the likelihood of the tracks given the set of fea-

tures [189]. Let the set of B features $\mathcal{Z} = \{\mathbf{z}_b\}_{b=1}^{B}$ be gathered from the video sequence and $\mathcal{T}^*$ be the set of track hypotheses obtained by associating features over time. The goal is to maximise the posterior probability of $\mathcal{T}^*$ given the set $\mathcal{Z}$ (Fig. 2.4b), that is

$$\mathcal{T} = \arg\max_{\mathcal{T}^*} p(\mathcal{T}^*|\mathcal{Z}) = \arg\max_{\mathcal{T}^*} p(\mathcal{Z}|\mathcal{T}^*)p(\mathcal{T}^*) = \arg\max_{\mathcal{T}^*} \prod_{b=1}^{B} p(\mathbf{z}_b|\mathcal{T}^*)p(\mathcal{T}^*), \tag{2.4}$$

where $\mathcal{T} = \{\mathsf{T}_a\}_{a=1}^{A}$ is the set of tracks and the likelihood probabilities are assumed to be conditionally independent given the hypothesis $\mathcal{T}^*$. Such maximisation can be calculated with optimal algorithms such as dynamic programming (e.g. Viterbi) [9,126,132,178]. Dynamic programming allows one to find the global optimal solution by decomposing the problem into subproblems in order to reduce the complexity. For example, target detections in a video sequence can be represented with nodes. The optimal association among all nodes can be found by applying the Viterbi algorithm every two consecutive frames and then by repeating the same operation for the other frames [178]. Alternatively, optimal solutions between each consecutive frame pair can be found through iterative methods that cycle through the sequence [36]. The iterative cycling method, similar to the Iterated Conditional Modes algorithm [23], can be used to update joint solutions of multiple variables in order to find stronger local optima, and the iterations continue until no further improvement are achieved. Two-frame optimal solutions are calculated by using 2D target locations with the Hungarian algorithm [92]. There are methods to iteratively compute optimal tracks using the complete set of features [36], and methods that reduce the complexity of the problem by pruning negligible hypotheses and by finding sub-optimal solutions in multiple steps [74, 103].

An alternative method is formulated with a cost-flow network [29, 74, 132]. Instead of using thresholds to link features [74], it is possible to use the algorithm for min-cost flow networks proposed by Goldberg [57]. The log-likelihood linking is calculated by taking into account size, position, appearance and time gap of the features by considering independence among them. Within such network formulation, each node is a detection and each flow through the network is interpreted as a track of a single target. The cost of the flow corresponds to the log-likelihood of the link hypothesis [132]. It is also possible to consider each node as a pair of detections with the advantage of adding high-order constraints (velocity) into the edges of the network [29]. These constant velocity constraints allow one to evaluate track smoothness over three consecutive frames instead of two [132], in order to highly penalise hypothesised tracks that locally have

sudden speed variations. In scenes, such as sport, where appearance features are less discriminative (people dressed the same) and where velocity variations are more frequent, additional related context features can also be employed within the network [107]. These features involves the distribution of the players over the court, relative distances among players within a certain radius, likely future location of the ball [133] and chasing links to detect if a player is marking another. Alternatively, features can be associated within a temporal window using Mean-Shift clustering [38] on the feature space [19]. For each cluster, which ideally represents a target, Principal Component Analysis (PCA) is applied and the features are associated by considering the direction of the principal components. PCA allows one to represent the local trend in the data distribution and measure the reliability of the associated features.

Final tracks can be obtained by tracklet association. On the one hand, this problem can be formulated as a joint problem of ranking and classification [103] by using HybridBoost, a combination of RankBoost [53] and AdaBoost [152]. The role of RankBoost is to build the tracklet affinity model considering relative preferences over any tracklet pairs as well as low values for those tracklet pairs that should not be associated. AdaBoost is composed of weak classifiers relying on a single type of feature for tracklet affinity measurements, such as appearance, motion or frame gap between a tracklet pair. On the other hand, the association can be performed with a network flow formulation (as described above) by considering each node as a tracklet and the edges as the links among tracklets. It is possible to constrain solutions to have a fixed number of trajectories, A, by pushing A units of flow between the source and sink nodes in the network [107].

An algorithm for optimal tracklet association (OLDAM) [94] uses a temporal shifting window for the online learning of discriminative appearance features. Positive samples are extracted within the same tracklet and collected for all the tracklets in a temporal window. Negative samples are collected by extracting features from tracklets not belonging to the same target and by taking into account their spatio-temporal properties. The model learning problem is formulated as a binary classification problem using AdaBoost. Affinity measurements of appearance features (i.e. colour and HOG) are adopted in AdaBoost to learn weak classifiers. The predicted confidence output of AdaBoost is combined with motion and time features in order to compute the link probability between tracklets. OLDAM has been further improved with PIRMPT [95], which includes a method to automatically select the most discriminative features from each tracklet by an

online learning method based on appearance descriptors. Such descriptors are used to create a target model for each tracklet and further employed to link consecutive tracklets [74]. Tracking improvements can be achieved with the Explicit Occlusion Model (EOM), which includes occlusion hypotheses in the tracking problem [189]. The EOM method generates a set of occlusion hypotheses and constraints and combines them with the input associations. This combination avoids errant associations due to large temporal gaps between the associated features.

In order to make the tracklet linking generic, methods should be independent of feature extractors, for example, by employing an optimisation process based on common affinity models along with social grouping behaviours [135]. The nonlinear equations used for the association can have terms approximated with Lagrange theory and solved using an iterative algorithm that employs the Hungarian algorithm and K-mean clustering [49].

2.4.4 Track initialisation and termination

Initialisation and termination of tracks are two important track management issues. The initialisation for causal trackers can be performed automatically, i.e. a new track starts when new features are available and are not associated to any of the existing tracks. The initialisation for non-causal trackers can be performed as for the causal trackers or with an implicit modality, i.e. when the initial location is associated to a track obtained as the optimal solution. An alternative is manual track initialisation, used for example in tag-and-track applications [12, 90].

Tracking methods performing batch association of features or tracklets [36, 94, 95, 181, 189] implicitly initialise and terminate tracks. In fact, when the optimal track solution is computed, the start and the end of each track are implicitly encoded into the solution. Instead, methods performing sequential localisation need criteria for track initialisation and termination.

Trackers such as Track-Before-Detect based on particle filter [142] perform joint detection and tracking of targets without relying on any external mechanism for initialisation or termination. The initialisation and termination of tracks are embedded in the filter and modelled using a Markov chain [130], where the number of states corresponds to the number of targets in the scene. Alternatively, if the target states are represented as a collection of random variables that create a finite-set-valued state modelled with a multi-Bernoulli RFS [110], the tracker can handle track initialisation and termination by relying on probabilities of target appearance and disappearance [68]. The RFS framework can handle a time-varying number of targets as well as missing and noisy features by employing, for example, the Probability Hypothesis Density

(PHD) filter [113].

External mechanisms for track initialisation or termination can be based on the extracted features [10, 27, 128, 172]. New tracks are initialised when none of the running trackers are associated to the localised targets [10,128]. This process can be enhanced when multiple features are generated along the image borders, which is an indication on new incoming targets in the scene [27]. The termination of a track occurs when the tracker is unable to validate the features for a number of consecutive frames. Also, ad-hoc methods for initialising and terminating tracks can be used by implementing clustering strategies on the extracted features and comparing the number of clusters in the current frame with those in the previous frame [18].

2.4.5 Scene contextual information

Context is exploited to distinguish targets from clutter [109] and to improve initialisation and termination of tracks [181]. For example, background information can be used to enhance the separability between target features and background features [159], or object-level information can be used to model spatio-temporal relationships in order to improve tracking in indoor scenarios [87, 104]. Scene contextual information includes the knowledge of the scene background, occlusion areas, entry/exit regions, and dynamic textures (Fig. 2.5).

Contextual information can be extracted by learning the environment from user annotations or automatically from the output of the tracker. *User annotation* of entry/exit regions [181] may be required in order to provide the tracker with reliable contextual information. For example, detections of targets located in manually selected entry regions can be used to initialise tracks [27]. Alternatively, entry/exit regions, typical paths and stopping regions can be *automatically* extracted from long-term tracks [82, 114] or tracklets [181, 192]. In unstructured scenes, contextual information can be used to perform online learning of motion maps [181]. A motion map can be constructed by relying on entry/exit regions of the scene and by using motion patterns gathered from tracks. Entry/exit regions are used to draw likely target paths when reliable target features are collected. The learning of non-linear motion patterns is used to enhance the diversity among different track hypotheses, to improve the affinity estimations among extracted features and to build robust appearance models [181]. In structured scenes, scene context can be incorporated by automatically learning floor fields [12], which model directions of people on dominant paths and towards preferred exit regions, and to improve the motion prediction in these regions. Finally, with an *interactive* learning environment, models can be learned for cluttered areas and

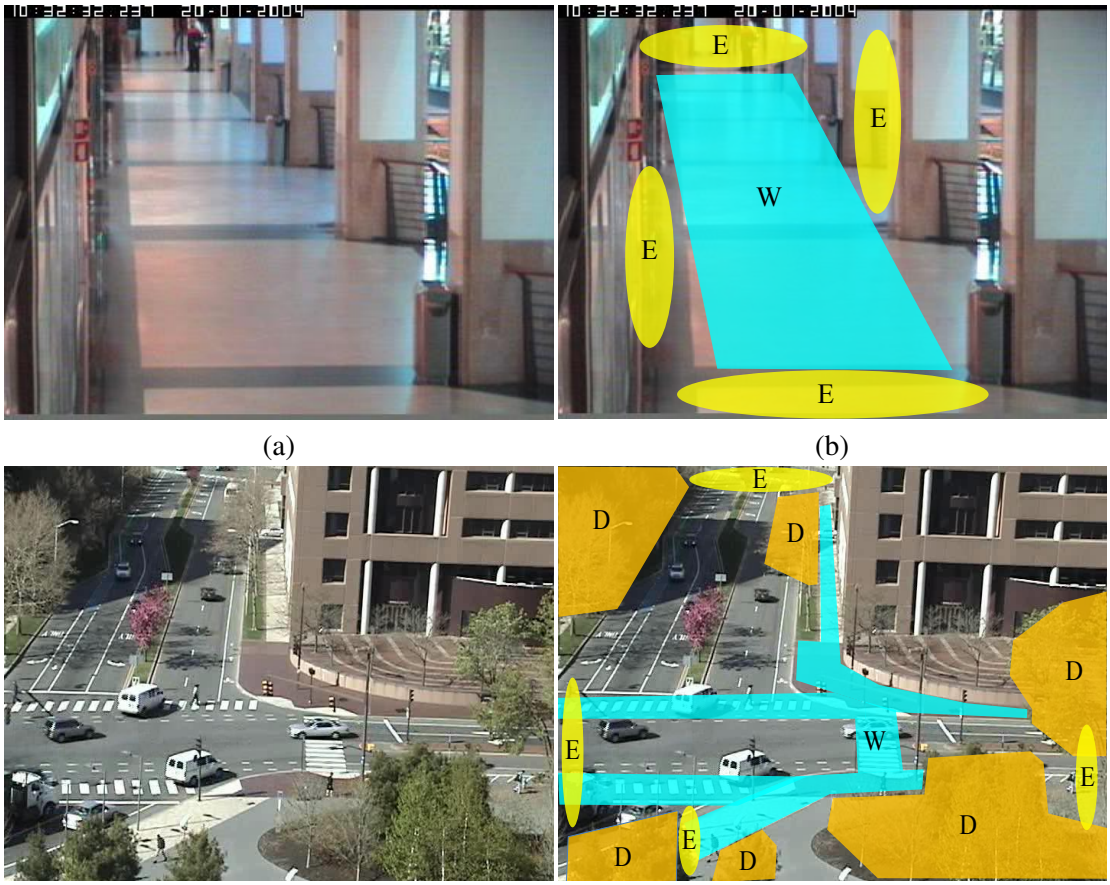


Figure 2.5: Examples of scene contextual information. Multi-target tracking can be enhanced by exploiting knowledge about the scene layout and objects such as trees that may occlude targets or may generate dynamic textures. Typical walking paths can also be used to narrow the search of human targets. Entry and exit regions can help track initialisation and track termination. (Key: D: Dynamic textures; W: Walking path; E: Entry/exit regions.)

for initialisation areas. The clutter model improves the tracker's capability to discard noisy measurements. The initialisation model can reduce the delay of track initialisation in locations where targets are likely to appear [109].

Scene contextual information can also be modelled and used to improve tracking accuracy when linking tracklets [74, 181]. For example, the scene model (i.e. entry/exit regions and static occluders) projected on the ground plane with homography from the image plane [59] can be used to reduce track fragmentation and prevent identity switches of linked tracklets. Long-range trajectory association is performed using an Expectation-Maximisation (EM) algorithm. The E-step estimates the scene model in terms of entry/exit regions with a Bayesian inference. These regions are then used to specify initialisation and termination of each tracklet. The M-step links tracklets using the information from the E-step and long tracks are obtained through the Hungarian algo-

rithm [74, 181, 188]. The assignment matrix used by the Hungarian algorithm is formulated as a MAP problem, relying on link probabilities calculated with associated detection responses.

2.5 Multi-target tracking evaluation measures

The evaluation of multi-target tracking methods generally quantify the discrepancy between estimated and ground-truth target regions [81, 124]. Unlike single-target tracking evaluation [99, 124], multi-target tracking evaluation requires solving the assignment problem to establish the associations between estimated and ground-truth targets [24, 28, 81, 143, 187], which is different from single-target tracking evaluation where only one track estimation has to be assessed [99, 123, 124]. The association can be computed using position only (point-based assignment) or region information as well (region-based assignment), and can be solved at frame level [20] or at sequence level [143]. *Point-based assignment* is based on distance minimisation between estimated and ground-truth tracks [24, 143], whereas *region-based assignment* can be based on the amount of overlap between estimated and ground-truth target regions [81, 187] or on their *coincidence* [28]. Coincidence occurs when the centroid of an estimated target lies within the region of a ground-truth target.

In the following sections we analyse state-of-the-art multi-target tracking evaluation measures by classifying them into four categories: Application-Dependent Assignment-based (ADA) evaluation, Point-based Assignment and Position-based (PAP) evaluation, Region-based Assignment and Position-based (RAP) evaluation and Region-based Assignment and Size-based (RAS) evaluation.

2.5.1 Application-Dependent Assignment-based measures

ADA measures can use position-based and region-based assignment, and provide tracking evaluation by taking into account target-size changes and target position only. The association between target estimation and ground truth is performed on a frame-by-frame basis. Examples of ADA measures include Precision (P), Recall (R) or F-Score (F) [141].

P and R use True Positive (TP), False Positive (FP) and False Negative (FN) track matches to compute the tracking accuracy, whereas F averages, with a weighting factor, P and R in order to obtain a single score value. TP, FP and FN can be defined in different ways based on the application. On the one end, they can be defined using position information only [105], where

a TP (FP) track match occurs when the distance between estimated and ground-truth states is below (above) a certain threshold distance, similar to the cut-off distance used in Optimal Sub-Pattern Assignment metric [143] (see Sec. 2.5.2). On the other end, P and R can be defined using region information, where a TP (FP) track match occurs when the overlapping region between estimated and ground-truth states is above (below) a certain threshold overlap, similar to τ_{TP} used in Multiple Object Tracking Accuracy [20,27] (see Sec. 2.5.4). In general P is calculated as

$$P = \frac{TP}{TP+FP}, \tag{2.5}$$

and Recall as

$$R = \frac{TP}{TP+FN}, \tag{2.6}$$

where TP is the number of true positive tracks of the sequence, FP the number of false positive tracks and FN the number of false negative tracks. F is then often calculated as

$$F = 2\frac{P \cdot R}{P+R}. \tag{2.7}$$

These measures are often used in the literature due to their generic formulation, and they can be employed in different fields such as information retrieval. Due to this, parameters have to be set and justified dependent on the application to evaluate.

2.5.2 Position-based Assignment and Position-based measures

PAP measures use a point-based assignment and evaluate target position only, without considering temporal size-changes. Examples of PAP measures include Object Tracking Error (OTE), the Wasserstein's distance-based metric, the Optimal Sub-Pattern Assignment (OSPA) metric, Tracker Detection Rate (TRDR), False Alarm Rate (FAR), Track Detection Rate (TDR) and Track Fragmentation (TF).

OTE [24] computes the average positional distance between each ground-truth and estimated track pair t. The association between the estimated and ground-truth tracks is performed by minimising the average Euclidean distance across the frame when they both exist [154]. For each t, OTE_t is calculated as

$$\text{OTE}_t = \frac{1}{K_t} \sum_{k=k_{t,s}}^{k_{t,e}} \sqrt{(x_{t,k} - {}^{g}x_{t,k})^2 + (y_{t,k} - {}^{g}y_{t,k})^2}, \tag{2.8}$$

where $K_t = k_{t,e} - k_{t,s}$ is the number of frames that are common in both ground-truth and estimated tracks and $k_{t,s}$ and $k_{t,e}$ denote the initial and final frame numbers, respectively, of the pair t.

The Wasserstein's distance-based metric [66], $W_p(\mathcal{T}_k, {}^g\mathcal{T}_k)$, computes the *p-norm* between estimated and ground-truth tracks as

$$W_p(\mathcal{T}_k, {}^g\mathcal{T}_k) = \min_{\mathbf{C}} \left(\sum_{a=1}^{u_k} \sum_{d=1}^{{}^g u_k} C_{a,d}^k d(x_{a,k}, {}^g x_{d,k})^p \right)^{1/p}, \tag{2.9}$$

where $d(\cdot)^p$ denotes the *p-norm* ($p \in [1,\infty)$) and $\mathbf{C}$ is the transportation matrix defining the association costs among all possible pairs of estimated and ground-truth tracks at frame k. The associations that minimise the overall cost determine the error value and are calculated by using the Hungarian or Munkres algorithms [92, 122].

Similarly to the Wasserstein's distance, OSPA [143, 153] defines the tracking error as

$$\mathcal{D}_{p,c}(\mathcal{T}_k, {}^g\mathcal{T}_k) = \left[\frac{1}{max(u_k, {}^g u_k)} \left(\min_{\pi \in \Pi_{u_k}} \sum_{d=1}^{{}^g u_k} \left(D_c({}^g\hat{\mathbf{x}}'_{d,k}, \hat{\mathbf{x}}'_{\pi(d),k}) \right)^p + |u_k - {}^g u_k| \cdot c^p \right) \right]^{1/p}, \tag{2.10}$$

where Π_{u_k} represents the set of permutations each containing ${}^g u_k$ elements taken from $\{1, 2, \ldots, u_k\}$, $\pi(d)$ indexes the elements with each permuted set Π_{u_k}, $D_c({}^g\hat{\mathbf{x}}', \hat{\mathbf{x}}') = \min(c, D({}^g\hat{\mathbf{x}}', \hat{\mathbf{x}}'))$ is the cut-off distance (defined below) between two states with $c>0$ representing the cut-off parameter and $p \in [1,\infty)$ is the order parameter of the OSPA-based metric. $D({}^g\hat{\mathbf{x}}', \hat{\mathbf{x}}')$ denotes the base distance that quantifies the discrepancy between generic estimated and ground-truth states, and includes localisation and labelling errors [143]:

$$D({}^g\hat{\mathbf{x}}', \hat{\mathbf{x}}') = \left(||{}^g\hat{\mathbf{x}}' - \hat{\mathbf{x}}'||_{p'} + \alpha^{p'} \bar{\delta}[{}^g\xi_d, \xi_a] \right)^{1/p'}, \tag{2.11}$$

where $\bar{\delta}[{}^g\xi_d, \xi_a]$ is the complement of the Kronecker delta such that $\bar{\delta}[{}^g\xi_d, \xi_a] = 0$ if ${}^g\xi_d = \xi_a$ and $\bar{\delta}[{}^g\xi_d, \xi_a] = 1$ if ${}^g\xi_d \neq \xi_a$, and $\alpha \in [0, c]$ is the penalty applied to the labelling error if the frame-level assignment (determined as a result of the minimisation in Eq. 2.10) does not correspond to the global assignment of tracks computed a priori. The global assignment is determined based on the minimisation of the average distance between estimated and ground-truth tracks [54, 143]. $p' \in [1,\infty)$ denotes the order parameter of the base distance. Typically, $p = p' = 1$ [143]. Unlike OTE and the Wasserstein's distance-based metric, OSPA incorporates the cardinality error in the evaluation procedure, which otherwise would not be taken into account by the minimisation term

of the distance error in Eq. 2.10.

TRDR, FAR and TDR [24] evaluate the accuracy using true positives ($\widehat{TP}_k$) and false positives ($\widehat{FP}_k$) at each frame k determined with the coincidence criterion. Although these measures use target-size information in the evaluation, they do not evaluate target-size changes over time, hence we consider them as PAP. For TRDR, FAR and TDR, the assignment between estimated and ground-truth tracks is solved as for OTE. In particular, TRDR quantifies the overall performance as

$$\text{TRDR}_k = \frac{\widehat{TP}_k}{{}^{g}u_k}, \tag{2.12}$$

which is the ratio of the number of correctly-tracked targets (true positives) to the number of ground-truth targets at k. An estimation is considered a true positive if the centroid of the ground-truth bounding box lies within the estimated bounding box (coincides). If none of the centroids of ground-truth bounding boxes coincides with an estimated bounding box, the estimation is considered a false positive.

FAR quantifies tracking performance as

$$\text{FAR}_k = \frac{\widehat{FP}_k}{\widehat{TP}_k + \widehat{FP}_k}, \tag{2.13}$$

which is the ratio of the number of incorrectly-tracked targets (false positives) to the sum of correctly- and incorrectly-tracked targets.

TDR quantifies the tracking performance at track level as

$$\text{TDR}_a = \frac{\widehat{TP}_a}{K_a}, \tag{2.14}$$

which is the ratio between the number $\widehat{TP}_a$ of true positive targets in the estimated track a and the number K_a of frames where the corresponding ground-truth track exists.

The evaluation of the consistency of the IDs of targets is provided in the form of TF [24], where the number of ID changes with respect to the ground-truth track is measured as the number of times a ground-truth track is associated with different estimated tracks. The association between estimated and ground-truth tracks is determined as for OTE.

2.5.3 Region-based Assignment and Position-based measures

RAP measures use a region-based assignment and provide a position-based evaluation. Examples of RAP measures include $\overline{TP}$ track matches, $\overline{FP}$ track matches and $\overline{FN}$ track matches. Their formulation is different from that described in Sec. 2.5.2.

The computation of $\overline{TP}$, $\overline{FP}$ and $\overline{FN}$ [28] is based on the spatial and temporal overlaps between estimated and ground-truth tracks and involves the computation of an implicit assignment. If the estimated track overlaps any ground-truth track both spatially and temporally, the estimation is considered a $\overline{TP}$ track match. A spatial overlap is achieved in a frame when the centroid of the estimated track coincides with the corresponding bounding box of the ground-truth track. At track level, it is measured for each ground-truth track as the percentage of frames having coincidence between estimated and ground-truth bounding boxes.

The temporal overlap $\bar{O}_{TP}$ for the case of $\overline{TP}$ between the ground-truth track d and the associated estimated track a is defined as

$$\bar{O}_{TP} = \frac{\mathcal{N}_{d,a}^{ov}}{K_d}, \tag{2.15}$$

where $\mathcal{N}_{d,a}^{ov}$ is the number of frames when ground-truth track d and estimated track a both exist. If the spatial or temporal overlap of the estimated track with any ground-truth track is smaller than a threshold $\tau_{\overline{FP}}$, the estimation is considered to be a $\overline{FP}$ track match. For a $\widehat{FP}$ match, the temporal overlap, $\bar{O}_{FP}$, between the estimated track a and the corresponding ground-truth track d is defined as

$$\bar{O}_{FP} = \frac{\mathcal{N}_{d,a}^{ov}}{K_a}. \tag{2.16}$$

Given all estimated tracks, if the spatial or temporal overlap of the ground-truth track with any estimated track is smaller than a threshold $\tau_{\overline{FN}}$ (different from $\tau_{\overline{FP}}$), the estimation is considered a $\overline{FN}$ match.

2.5.4 Region-based Assignment and Size-based measures

RAS measures use a region-based assignment and provide tracking evaluation that accounts for target-size changes over time (size-based evaluation). Examples of RAS measures include Correct Detected Track (CDT), False Alarm Track (FAT), Track Detection Failure (TDF), Multiple Object Tracking Precision (MOTP), Multiple Object Detection Accuracy (MODA), Normalised

MODA (N-MODA), Multiple Object Tracking Accuracy (MOTA) and ID changes (IDC).

CDT, FAT and TDF [187] are conceptually similar to $\overline{TP}$, $\overline{FP}$ and $\overline{FN}$ tracks [28]. Although they include the variations of target sizes in the evaluation, they do not individually evaluate the cardinality error. Differently from the coincidence-based methods [28], the spatial overlap is measured as the number of common pixels between estimated and ground-truth bounding boxes. For MOTP, MODA, Normalised MODA and MOTA, a one-to-one assignment is achieved at frame level between estimated and ground-truth tracks based on the maximisation of spatial overlap values (computed as for CDT, FAT and TDF) between pairs using, for example, the Hungarian algorithm [81, 92].

MOTP [81] is a spatio-temporal measure that computes the amount of overlap between estimated and ground-truth tracks:

$$\text{MOTP} = \frac{\sum_{t=1}^{A'} \sum_{k=k_{t,s}}^{k_{t,e}} \frac{|{}^{g}S_{t,k} \cap S_{t,k}|}{|{}^{g}S_{t,k} \cup S_{t,k}|}}{\sum_{k=1}^{K} A'_k}, \tag{2.17}$$

where A' is the number of associated estimated and ground-truth track pairs in the sequence, $k_{t,s}$ and $k_{t,e}$ denote the initial and final frame, respectively, of the common time interval of one associated track pair t. $|{}^{g}S_{t,k} \cap S_{t,k}|$ is the number of common pixels in ${}^{g}S_{t,k}$ and $S_{t,k}$, $|{}^{g}S_{t,k} \cup S_{t,k}|$ is the number of pixels in ${}^{g}S_{t,k} \cup S_{t,k}$, and A'_k is the number of associated estimated and ground-truth target pairs at frame k. The pairs with an overlap greater than a fixed threshold value τ_{TP} are considered in the evaluation procedure.

MODA_k [81] computes tracking performance at frame k by combining the information about the number of false negative estimations FN_k and the number of false positive estimations FP_k:

$$\text{MODA}_k = 1 - \frac{c_1\, FN_k + c_2\, FP_k}{{}^{g}u_k}, \tag{2.18}$$

where c_1 and c_2 are fixed a priori. FP_k and FN_k are determined by comparing the amount of overlap between estimated and corresponding ground-truth targets with the threshold τ_{TP}. Note that MODA is not numerically lower bounded. For example, let $c_1 = c_2 = 1$, $FN_k = 2$, $FP_k = 6$ and ${}^{g}u_k = 6$; hence $\text{MODA}_k = -0.33$. As FN_k and/or FP_k increase, MODA_k keeps decreasing without lower bound. A sequence-level formulation of MODA, the Normalised MODA (N-

MODA) [81], is defined as

$$\text{N-MODA} = 1 - \frac{\sum_{k=1}^{K}(c_1\, FN_k + c_2\, FP_k)}{\sum_{k=1}^{K} {}^{g}u_k}. \tag{2.19}$$

MOTA [81] is a sequence-level measure that evaluates tracking performance by including also the information about the number of ID switches (IDS_k) in each frame, in addition to FN_k and FP_k. FN_k, FP_k and IDS_k contributions are determined by manually setting the corresponding three application-dependent parameters, c_1, c_2 and c_3, respectively. The contributions are accumulated across the sequence and normalised as follows:

$$\text{MOTA} = 1 - \frac{\sum_{k=1}^{K}(c_1\, FN_k + c_2\, FP_k + c_3\, IDS_k)}{\sum_{k=1}^{K} {}^{g}u_k}, \tag{2.20}$$

where FP_k and FN_k are determined as in MODA and, as with MODA, it is not numerically lower bounded. For example, let $c_1 = c_2 = c_3 = 1$ and $k = 1,2$; at $k = 1$, $FN_1 = 0, FP_1 = 2, IDS_1 = 0$, ${}^{g}u_1 = 3$; at $k = 2$, $FN_2 = 0$, $FP_2 = 5$, $IDS_2 = 2$, ${}^{g}u_2 = 3$; hence, MOTA $= -0.50$.

IDC [187] counts the number of ID changes corresponding to all ground-truth tracks. At each frame, each estimated bounding box is assigned to the ground-truth bounding box with an overlap larger than a predefined threshold. An ID change occurs when the amount of overlap between an estimated and ground-truth track goes below the threshold.

2.6 Discussion

In this chapter we presented state-of-the-art multi-target trackers, discussing their major processing stages, which include detection algorithms, prediction models, localisation and association methods, and techniques for track initialisation and termination. We discussed how contextual information can be employed to improve tracking performance. We also observed that some algorithms explicitly define locations in the scene where the tracking should be initialised (e.g. [27]). Table 2.2 summarises state-of-the-art methods of multi-target trackers. We aim at formulating an online and buffered multi-target tracking algorithm where the initialisation and termination of tracks will be implicitly performed by the algorithm in any location of the scene. Unlike [90], we do not use any trained or tailored motion model (prediction), so that the algorithms will be flexible for different scenarios such as surveillance and biology. In the case of surveillance scenarios we use a postprocessing stage which embeds the knowledge that the tracked objects are people

Table 2.2: Taxonomic summary of multi-target trackers. Key: Ref: Reference; TI: Track initialisation; TT: Track termination; BS: Background subtraction; RGB: Red Green Blue colorspace; HSV: Hue Saturation Value colorspace; HOG: Histogram of Oriented Gradients; ISM: Implicit Shape Model; WFD: Weighted Frame Difference; SIFT: Scale-Invariant Feature Transform; I: Implicit; A: Automatic; T&T: Tag and track; '-': no information provided.

Ref.	Feature extraction				Motion model		Sequential localisation	Batch association	TI	TT
	BS	Colour	Shape	Texture	Pre-learned	Fixed				
[74]		RGB	Edgelets			✓	✓	✓	I	I
[189]		RGB	Edgelets			✓		✓	I	I
[103]		RGB	Edgelets			✓		✓	I	I
[94]		RGB	Edgelets+HOG	Covariance matrix		✓	✓	✓	I	I
[95]		RGB	Edgelets+HOG	Covariance matrix		✓	✓	✓	I	I
[181]		RGB	Edgelets+HOG	Covariance matrix	✓		✓	✓	I	I
[36]						✓		✓	I	I
[135]		HSV				✓		✓	I	I
[190]	Gaussian	RGB				✓	✓		A	A
[128]		HSV				✓	✓		A	-
[43]		RGB				✓	✓		A	A
[18]	WFD					✓	✓		A	A
[20]			HOG			✓	✓	✓	I	I
[145]			HOG		✓		✓		T&T	-
[144]		-			✓		✓		T&T	-
[12]		-			✓		✓		T&T	-
[90]			Gradient		✓		✓		T&T	-
[184]		RGB	Elliptical model	SIFT		✓	✓		A	-
[27]			HOG/ISM			✓	✓		A	A
[65]	-	RGB				✓	✓		A	-
[10]		HSV				✓	✓		A	A
[159]	Gaussian+Vessel					✓	✓		A	A
[68]		HSV				✓	✓		A	A
[J2]						✓	✓		I	I

and the colour feature is used to distinguish targets. In the case of biology, the colour cannot be used as a distinguishing feature since targets have the same appearance, hence only location information will be used and more effort will be put into dealing with the association of noisy data.

Moreover, we presented the state-of-the-art procedures for the assessment of multi-target tracking performance in the case of extended targets. We analysed measures for application-dependent assignment measures, point-based assignment and position-based measures, region-based assignment and position-based measures, and region-based assignment and size-based measures. Table 2.3 compares the state-of-the-art multi-target tracking evaluation measures. Existing frame-level measures do not take into account the evaluation of target-size changes [24, 66, 143] and require presetting application-dependent parameters [81, 141, 143]. Additionally, frame-level measures ignore the cardinality error [24, 66]. Sequence-level measures do not evaluate target-size changes (e.g. OTE, TDR [24] and the measures presented in [28]) and use application-dependent thresholds (e.g. P, R, MOTA, MOTP [81] and the measures presented in [187]). These measures aim only to evaluate the accuracy while not considering the cardinality error [24, 28, 81, 187]. Existing sequence-level measures are generally not employed to

Table 2.3: Comparison of multi-target tracking evaluation measures. Key: PI: parameter independence; SE: size-change evaluation; APS: assignment problem solution; PB: point-based; RB: region-based; FL: frame-level measure; SL: sequence-level measure; AE: accuracy error; CE: cardinality error; Prop.: proposed; TP_m: TP matches; FP_m: FP matches; FN_m: FN matches.

Ref.	Measure	PI	SE	APS	Type	AE	CE
[141]	Precision		✓	PB, RB	FL, SL	✓	✓
[141]	Recall		✓	PB, RB	FL, SL	✓	✓
[141]	F-score		✓	PB, RB	FL, SL	✓	✓
[143]	OSPA			PB	FL	✓	✓
[66]	$W_p(\cdot)$	✓		PB	FL	✓	
[24]	OTE	✓		PB	SL	✓	
[24]	TRDR	✓		PB	FL	✓	
[24]	FAR	✓		PB	FL	✓	
[24]	TDR	✓		PB	SL	✓	
[24]	TF	✓		PB	SL		
[28]	TP_m			RB	SL	✓	
[28]	FP_m			RB	SL	✓	
[28]	FN_m			RB	SL	✓	
[187]	CDT		✓	RB	SL	✓	
[187]	FAT		✓	RB	SL	✓	
[187]	TDF		✓	RB	SL	✓	
[187]	IDC		✓	RB	SL		
[81]	MODA		✓	RB	FL	✓	✓
[81]	N-MODA		✓	RB	SL	✓	✓
[81]	MOTA		✓	RB	SL	✓	✓
[81]	MOTP		✓	RB	SL	✓	
[J1]	METE	✓	✓	RB	FL	✓	✓
[J1]	MELT	✓	✓	RB	SL	✓	
[J1]	NIDC	✓	✓	RB	SL		

analyse tracking at varying accuracy levels, which would be desirable and useful to determine the suitability of trackers for different applications or scenarios. ID-change evaluation measures simply incorporate the information about the total number of ID changes or switches in the sequence [24, 81, 187]; however it would be desirable to evaluate ID changes relative to the track duration.

Interesting open challenges in multi-target tracking may also include the effective extension of feature selection for target-background separability from offline [160] to on-line approaches [148], defining motion models that are flexible to deal with different dynamics of a scene [145], and predicting tracking failures by identifying image regions where trackers are likely to fail [79]. These failures can be detected by employing interaction models based on track information [83] and potentially solved by strengthening the trackers with methods for self-tuning parameters [100] (e.g. a resampling strategy for a particle filter [142]). Removing the dependence of user interaction is also desirable to make the environment learning stage flexible to context changes [82, 114] and independent from user feedback [109].

Chapter 3

Multi-target tracking on confidence maps

3.1 Introduction

Tracking methods employ target localisations as measurements, either directly as unthresholded data (confidence maps) [27, 84, 96, 159] or as binary maps (target/non-target information) obtained by thresholding the confidence values [20, 74, 93, 101]. Target localisations can be gathered from sensors (e.g. laser, sonar, camera) that provide multiple measurements per target and carry information in the form of confidence values over space (Fig. 3.1). These confidence values are affected by different types of *noise* on background areas and/or on the targets themselves, thus resulting in inaccurate (noisy) position estimations. Although the latter strategy is the most commonly used, relevant data may be lost with this process. Tracking-by-detection methods [27] perform target-tracker association, and initialisation and termination of tracks with greedy algorithms. Track-before-detect (TBD) methods perform tracking of targets using unthresholded data [142] and target-tracker association is implicitly computed by the tracker. TBD is a Bayesian filter, generally built on the concept of particle filters, and commonly used for radar tracking [30, 142]. In fact tracking is performed on noisy confidence values and the targets are assumed to be point targets. Initialisation and termination of tracks are performed by the tracker using target *birth and death* models.

In this chapter, we propose a multi-target tracker based on TBD [142] and applied to confidence maps[1]. The confidence maps are assumed to be given. To enable multi-target tracking, we

[1]The work in this chapter appears in [J2].

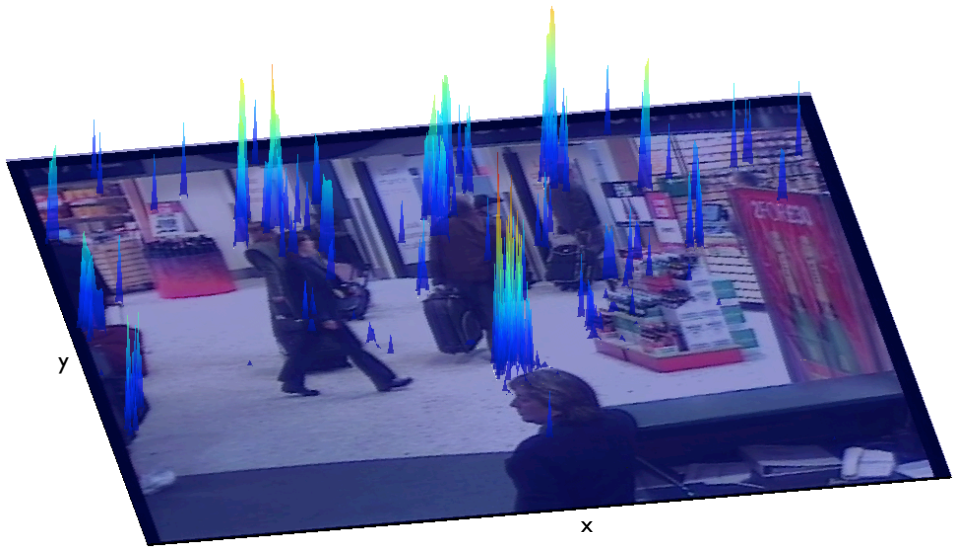


Figure 3.1: Sample confidence map that we use as input (observation) to simultaneously track multiple objects. In this example, the confidence map is obtained with a head localisation method based on [44].

develop a method where target identities (IDs) are assigned based on Mean-Shift clustering [39] and Gaussian Mixture Models (GMM) [49]. The birth and death of targets are modelled with a Markov Random Field (MRF) [87].

The chapter is organised as follows. In Sec. 3.2, we describe how the Bayesian estimation is performed. In Sec. 3.2.1 we define the confidence map and the TBD framework. Section 3.2.2 explains the inclusion of the multi-target identity into the Bayesian estimation and Sec. 3.2.3 describes how the Monte Carlo estimation is performed. The ID management via MRF is explained in Sec. 3.2.4. Section 3.3 and 3.4 give an example of likelihood modelling and postprocessing, respectively, in the case of extended targets (i.e. people), and in Sec. 3.5.3 we validate the performance of each part of the method. In Sec. 3.6, we summarise the achievement of the proposed multi-target tracker.

3.2 Bayesian estimation

3.2.1 Confidence maps and track-before-detect

Let a confidence map $\mathfrak{M}$ provide the information on the estimated position of targets through spatially-localised confidence values (Fig. 3.1). The ideal representation of the target position on a confidence map is a Dirac delta (a point target), with maximum confidence. In practice, such Dirac delta is a spread function centred in the target position and affecting neighbouring pixels.

Let the state vector $\mathbf{x}_k \in \mathfrak{X}$, where $\mathfrak{X}$ is the state space, be defined as

$$\mathbf{x}_k = [x_k \; \dot{x}_k \; y_k \; \dot{y}_k \; I_k]^T, \tag{3.1}$$

where (x_k, y_k) is the position, $(\dot{x}_k, \dot{y}_k)$ the velocity, I_k the confidence of the point target and T is the symbol for the transposed matrix. TBD is a discrete-time system that observes multiple moving targets on a 2D image. The evolution of the targets at each frame k is described by a discrete and linear Gaussian model [142]:

$$\mathbf{x}_k = F_{\mathbf{x}}\mathbf{x}_{k-1} + \omega_{k-1}. \tag{3.2}$$

The transition matrix $F_{\mathbf{x}}$ describes the evolution of the target at a constant velocity:

$$F_{\mathbf{x}} = \begin{bmatrix} 1 & \mathcal{K} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \mathcal{K} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.3}$$

where $\mathcal{K}$ denotes the sampling period. The noise of this evolution is normally distributed and defined as $\omega_{k-1} \sim \mathcal{N}(0, Q)$, with variance

$$Q = \begin{bmatrix} \frac{q_1}{3}\mathcal{K}^3 & \frac{q_1}{2}\mathcal{K}^2 & 0 & 0 & 0 \\ \frac{q_1}{2}\mathcal{K}^2 & q_1\mathcal{K} & 0 & 0 & 0 \\ 0 & 0 & \frac{q_1}{3}\mathcal{K}^3 & \frac{q_1}{2}\mathcal{K}^2 & 0 \\ 0 & 0 & \frac{q_1}{2}\mathcal{K}^2 & q_1\mathcal{K} & 0 \\ 0 & 0 & 0 & 0 & q_2\mathcal{K} \end{bmatrix}, \tag{3.4}$$

where q_1 and q_2 are noise levels in target motion and confidence, respectively.

Let the spread function of the estimated positions of targets (over the 2D image) be modelled as

$$h_k^{(i,j)}(\mathbf{x}_k) = I_k \exp\left\{-\frac{(i-x_k)^2 + (j-y_k)^2}{2\Sigma^2}\right\}, \tag{3.5}$$

where Σ is a known parameter that represents the amount of blurring (i.e. the spread of the confidence) and (i, j) is the pixel position.

The recursive Bayesian filtering involves the calculation of the *posterior* probability density function (pdf) $p(\mathbf{x}_k|\mathbf{Z}_k)$ of $\mathbf{x}_k$ given the observations up to frame k, $\mathbf{Z}_k = \{\mathcal{Z}_{k'}\}_{k'=1}^{k}$. The posterior is calculated in two steps: *prediction* and *update*. In the prediction step, the probability density function is calculated through a prior distribution, which determines the state evolution through the motion model. In the update step, when observation $\mathcal{Z}_k$ is available, the prediction is updated using the likelihood function. The posterior pdf is thus obtained with Bayesian recursion as

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \frac{p(\mathcal{Z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathcal{Z}_k|\mathbf{Z}_{k-1})}, \tag{3.6}$$

where $p(\mathcal{Z}_k|\mathbf{x}_k)$ is the *likelihood* function, $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$ is the prediction density and $p(\mathcal{Z}_k|\mathbf{Z}_{k-1})$ is a normalising constant calculated as

$$p(\mathcal{Z}_k|\mathbf{Z}_{k-1}) = \int_{\mathfrak{X}} p(\mathcal{Z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})d\mathbf{x}_k. \tag{3.7}$$

3.2.2 Multi-target identity

The framework for single-target tracking described in [142] (Ch. 11) includes in the state vector $\mathbf{x}_k$ an existence variable $E_k \in \{0,1\}$, where 0 (1) indicates a target's absence (presence). The global existence of a target over time (i.e. birth and target) is modelled with a two-state Markov chain. The further extension to multi-target [43] leads to the expansion of the state vector $\mathbf{x}_k$ and of the Markov chain proportionally to the number of the targets. Since the number of states of a Markov chain is fixed, the maximum number of targets must be known *a priori*. In addition to this, the Markov chain may not allow transitions from zero to two targets, and vice versa [43]. Alternatively, birth and death of multiple targets can be modelled either with greedy algorithms, where a target is declared born if the tracker receives its measurements within a certain period of time [125], or by a multi-Bernoulli distribution defining birth and death probabilities, and used to declare a target birth when the existence probability of a candidate target is larger than a certain threshold [68].

In order to be independent of the number of targets, we include in $\mathbf{x}_k$ the state variable ξ for representing the target identity (ID). IDs are represented by the set of random variables $\mathcal{L}_k = \{L_\xi\}_{\xi\in\Xi_k}$, where Ξ_k is the set of IDs at frame k and $p(L_\xi = \xi) = p(L_\xi)$. The IDs within Ξ_k at frame k depend on two factors: the IDs at $k-1$ and $\mathbf{x}_k$. Hence, we define $\Xi_k = g_{ID}(\Xi_{k-1}, \mathbf{x}_k)$, where $g_{ID}(\cdot)$ represents the function that (i) maintains target IDs; (ii) assigns new IDs to ap-

pearing targets (target births); and (iii) removes the IDs of targets that have disappeared (target deaths). Targets can move in any locations of the observed area and they might cross or move close to each other. By considering the IDs as random variables, we can assign the probability of having the corresponding ID to each target, such that

$$p(\mathbf{x}_k, L_\xi) = p(\mathbf{x}_k|L_\xi)p(L_\xi). \tag{3.8}$$

A target may spatially interact with other targets in its vicinity (neighbourhood). When targets are close to each other, there is uncertainty in assigning IDs. The main goal is to keep their identities separated and associated to the correct targets by maximising the probability of having their assigned ID. To this end, we take into account the selected targets with respect to the neighbouring ones in the calculation of the probability $p(L_\xi)\ \forall\ \xi$. The probability of a target having an ID depends only on the spatially close targets and, hence, the dependencies for the calculation of the probability follow the Markovian property. For this reason, to consider the state and its neighbourhood, we model the set $\mathcal{L}_k$ as a Markov Random Field (MRF). With such definition of $g_{ID}(\cdot)$ and $p(L_\xi)$, the proposed method of target birth and death lies between greedy and probabilistic methods.

Let us denote the neighbourhood of L_ξ as $\mathfrak{N}(\xi)$, where the Markovian property of L_ξ is defined via local conditions

$$p(L_\xi\,|\,\mathcal{L}_k \setminus \xi) = p(L_\xi\,|\,\mathfrak{N}(\xi)). \tag{3.9}$$

The information about the target identity within the state leads to the calculation of the likelihood and the prediction depending on the set $\mathcal{L}_k$, such that

$$p(\mathbf{x}_k, \mathcal{L}_k|\mathbf{Z}_k) = \frac{p(\mathcal{Z}_k|\mathbf{x}_k, \mathcal{L}_k)p(\mathbf{x}_k, \mathcal{L}_k|\mathbf{Z}_{k-1})}{p(\mathcal{Z}_k|\mathbf{Z}_{k-1})}. \tag{3.10}$$

By construction $\mathcal{L}_k$ is conditionally independent of the time and the observations $\mathbf{Z}_k$, and hence Eq. 3.10 can be rewritten as

$$p(\mathbf{x}_k, \mathcal{L}_k|\mathbf{Z}_k) = \frac{p(\mathcal{Z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})p(\mathcal{L}_k)}{p(\mathcal{Z}_k|\mathbf{Z}_{k-1})}, \tag{3.11}$$

where the prediction term $p(\mathbf{x}_k, \mathcal{L}_k|\mathbf{Z}_{k-1}) = p(\mathbf{x}_k|\mathbf{Z}_{k-1})p(\mathcal{L}_k|\mathbf{Z}_{k-1}) = p(\mathbf{x}_k|\mathbf{Z}_{k-1})p(\mathcal{L}_k)$ and the update term $p(\mathcal{Z}_k|\mathbf{x}_k, \mathcal{L}_k) = p(\mathcal{Z}_k|\mathbf{x}_k)$.

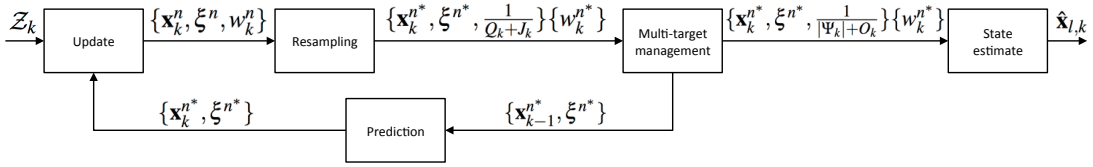


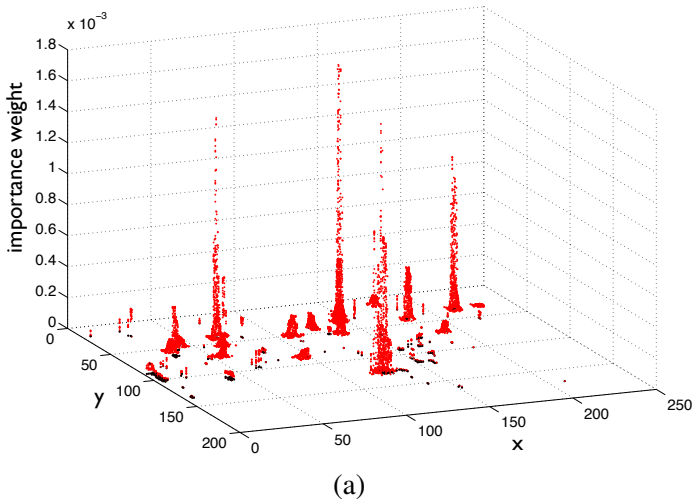
Figure 3.2: Block diagram of the proposed multi-target track-before-detect. The filter receives as input the confidence map ($\mathcal{Z}_k$) and draws the distribution of the target states using the Bayesian estimation with Monte Carlo approximation (particles). The weights of the particles are carried throughout the framework and used in the state estimation stage to find the target locations ($\hat{\mathbf{x}}_{l,k}$). The states marked with the superscript $*$ are generated after resampling. After the multi-target management stage, the weight distribution is uniform with respect to the number of the targets, O_k, at frame k.

3.2.3 Sequential Monte Carlo estimation

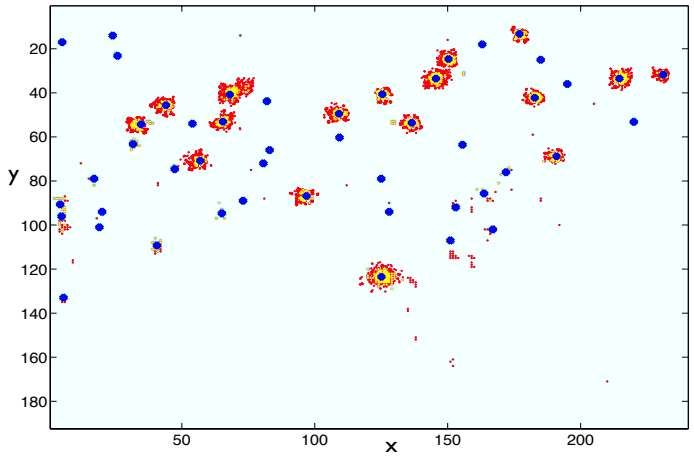
In order to make the Bayesian recursion of Eq. 3.11 computationally tractable, we use the Sequential Monte Carlo estimation to approximate the probability densities with a set of particles [142] (Fig. 3.2). The N particles used to describe the posterior $p(\mathbf{x}_k, \mathcal{L}_k|\mathbf{Z}_k)$ at frame k are denoted as $\{\mathbf{x}_k^n, \xi^n, w_k^n\}_{n=1}^N$, where w_k^n is the importance weight of the n^{th} particle.

In the prediction step there are two sets of particles: *existing* and *new-born* (Fig. 3.3(a)). The set of Q_k existing particles are drawn from the motion model of Eq. 3.2 and the set of J_k new-born particles are drawn from the proposal density $q_k(\mathbf{x}_k|\mathcal{Z}_k)$; both are chosen *a priori* and $N = Q_k + J_k$. The proposal density q_k distributes particles in $\mathcal{Z}_k$ proportionally to the confidence values of the input confidence map, thus resulting in a high concentration of particles in high-confidence regions. The proposal density for the velocity is uniformly distributed for both x and y components, e.g. for x, $q_k(\dot{x}_k) = \mathcal{U}[-\upsilon_{max}, \upsilon_{max}]$, where υ_{max} is the maximum target velocity and $\mathcal{U}[\cdot]$ indicates a uniform distribution in the interval defined within the squared brackets. The proposal density for the confidence component is $q_k(I_k) = \mathcal{U}[I_{min}, I_{max}]$, where I_{min} and I_{max} are the minimum and maximum confidence values, respectively. ξ is initialised with null value.

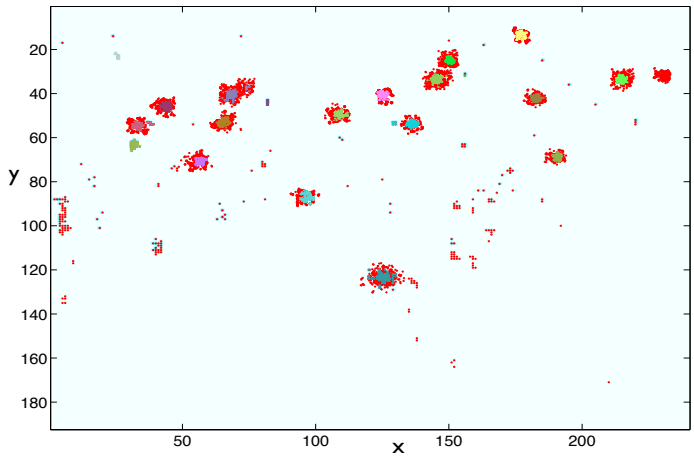
In the update step, the importance weights w_k^n are computed using the likelihood function. The likelihood modelling is performed in two steps: the extraction of confidence values of true and false target locations over time using ground-truth data from a training set (Sec. 3.3), and the fitting of a function on the collected data that minimises the distance between measured and predicted values [49]. The distribution of confidence values of true locations over time is referred to as *signal-plus-noise*, the distribution of confidence values of false locations as noise. The ideal case is with a Dirac delta in 0 for false locations (no noise) and a Dirac delta in 1 for true locations (clean signal). The likelihood is calculated as the ratio between the distribution



(a)



(b)



(c)

Figure 3.3: ID assignment, from prediction to state estimation. (a) Monte Carlo representation at the prediction step (red particles: existing particles propagated with the motion model from the previous time step; black particles: new-born particles). (b) Mean-Shift clustering result on the particles approximating the posterior distribution (blue markers: centroids of the clusters; yellow particles: particles kept in the resampling process). (c) Distribution of the particles with different IDs (colour-coded) superimposed on the actual observation (the confidence map).

of the target signal-plus-noise $p_{S+N}(\mathbf{z}_k^{(i,j)}|\mathbf{x}_k^n)$ and the distribution of the noise $p_N(\mathbf{z}_k^{(i,j)})$. In the former case, we use a Normal distribution and in the latter case a Pareto distribution [69].

Given the observation $\mathcal{Z}_k$, the likelihood $\ell(\mathbf{z}_k^{(i,j)}|\mathbf{x}_k^n)$ for the n^{th} particle at frame k and position (i,j) is calculated as

$$\ell(\mathbf{z}_k^{(i,j)}|\mathbf{x}_k^n) = \underset{i\in C_i(\mathbf{x}_k^n), j\in C_j(\mathbf{x}_k^n)}{\arg\max} \left\{ \frac{p_{S+N}(\mathbf{z}_k^{(i,j)}|\mathbf{x}_k^n)}{p_N(\mathbf{z}_k^{(i,j)})} \right\}, \tag{3.12}$$

where $C_i(\mathbf{x}_k^n)$ and $C_j(\mathbf{x}_k^n)$ are the set of pixels (the kernel) centred on pixel (i,j) and affected by the uncertainty mentioned in Sec. 3.2.1 during target localisation. The importance weights are finally calculated as

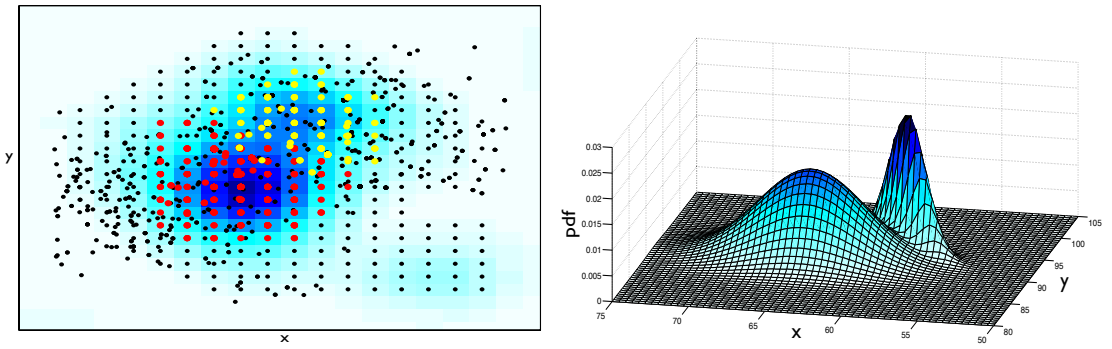
$$w_k^n = \frac{\ell(\mathbf{z}_k^{(i,j)}|\mathbf{x}_k^n)}{\sum_{n=1}^N \ell(\mathbf{z}_k|\mathbf{x}_k^n)} \cdot \frac{p(L_{\xi^n})}{\sum_{n=1}^N p(L_{\xi^n})}, \tag{3.13}$$

where $p(L_{\xi^n})$ is the ID probability of the n^{th} particle (Sec. 3.2.4). The importance weights approximate the updated posterior $p(\mathbf{x}_k, \mathcal{L}_k|\mathbf{Z}_k)$ whose modes represent the estimated state of the targets (Fig. 3.3(a)). To avoid the degeneracy problem [142], the particles are resampled using multinomial resampling. Resampling eliminates (duplicates) samples with low (high) importance weights. To retrieve the modes of the posterior distribution, we perform Mean-Shift (MS) clustering [39] using the position of the particles, i.e. $(x_k^n, y_k^n)\ \forall\ n$ (Fig. 3.3(b)), without any prior knowledge on the number of clusters or their shape, and with a fixed cluster size.

Let us define the size of the cluster as λ_Ψ and the set of clusters at frame k as $\Psi_k = \{\psi_r\}_{r=1}^{\mathcal{R}_k}$, with ψ_r the generic r^{th} cluster and $\mathcal{R}_k$ the number of clusters at k. At this stage, the function $g_{ID}(\cdot)$ introduced in Sec. 3.2.2 assigns a different ID to the particles belonging to different clusters at $k=1$. At $k>1$, if a cluster contains only new-born particles, they are all initialised with a new ID. Otherwise, the ID is assigned to the new-born particles with a method based on Gaussian Mixture Models (GMM), as explained in the next section.

3.2.4 ID management with Markov Random Fields

We address now the issues of managing multiple target identities in the presence of interactions, target births and target deaths. We use the random variable L_ξ as a contribution to the posterior distribution of Eq. 3.11 for penalising particles belonging to one target that either mix with particles of other targets or move far from their own target. Since the target measurement is spatially spread in a region (i.e. $C_i(\mathbf{x}_k^n)$ and $C_j(\mathbf{x}_k^n)$), particles belonging to a target are in turn spread over



(a) (b)

Figure 3.4: Example of Gaussian fitting on the particle states of two close targets. The GMM is used to assign IDs to new-born particles within a cluster containing targets with different IDs. (a) Red and yellow represent existing particles belonging to different targets, whereas black represents new-born particles. (b) Corresponding Gaussian mixture fitting on the particles.

the kernel (Fig. 3.3). Hence, when targets get close to each other, particles are likely to mix (Fig. 3.4(a)), thus creating a challenging situation to manage in order to separately maintain the identity of multiple targets.

To address this problem, let us characterise the set $\mathcal{L}_k$ and the joint probability distribution $p(\mathcal{L}_k)$. Since $\mathcal{L}_k$ is a MRF, in order to construct the joint distribution of $\mathcal{L}_k$ considering the Markovian property of Eq. 3.9, we employ the Gibbs distribution [87],

$$p(\mathcal{L}_k) = \frac{1}{D}\exp\{U(\mathcal{L}_k)\}, \tag{3.14}$$

where D is a normalisation factor and $U(\cdot)$ is the energy function

$$U(\mathcal{L}_k) = \sum_{\mathfrak{N}(\xi)\in\mathfrak{N}} V_{\mathfrak{N}(\xi)}(L_\xi), \tag{3.15}$$

where $\mathfrak{N}$ represents all the possible neighbourhoods in the state space and $V_{\mathfrak{N}(\xi)}$ is the potential function defined for the neighbourhood $\mathfrak{N}(\xi)$. Since a potential function is defined on a single neighbourhood, it ensures that it is possible to factorise the joint probability such that the conditionally independent variables, for instance from non-connected neighbourhoods, do not contribute to the same potential function.

Given a particle $\mathbf{x}_k^n$, the probability of ξ^n is $p(L_{\xi^n})$ and its neighbourhood $\mathfrak{N}(\xi^n)$ corresponds to the domain defined by the pixels affected by the blurring introduced during target localisation, i.e. $C_i(\mathbf{x}_k^n)$ and $C_j(\mathbf{x}_k^n)$ (Eq. 3.12).

The potential function of ξ^n at frame k associated to particle $\mathbf{x}_k^n$ is calculated as

$$V_{\mathfrak{N}(\xi^n)}(L_{\xi^n}) = V'_{\mathfrak{N}(\xi^n)}(L_{\xi^n}) + V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n}), \tag{3.16}$$

where $V'_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ evaluates the agreement of the ID as particle n with respect to the IDs in $\mathfrak{N}(\xi^n)$ and $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ evaluates the distance between the ID of particle n and the centre of mass of particles with the same ID of particle n. We define

$$V'_{\mathfrak{N}(\xi^n)}(L_{\xi^n}) = \exp\left\{-\alpha_1(1-\delta_k^n)\frac{\varkappa_k^n}{\rho}\right\}, \tag{3.17}$$

where $\varkappa_k^n$ quantifies the agreement of the IDs and α_1 regulates the strength of the agreement. For instance, a high value of α_1 leads to a low probability of having an ID when a particle is surrounded by particles with different IDs. Conversely low value of α_1 keeps the probability $p(L_{\xi^n})$ high when a particle is surrounded by particles with different IDs. ρ normalises the agreement value over the number of particles in the neighbourhood,

$$\varkappa_k^n = d_k^{\mathfrak{N}(\xi^n)} - a_k^{\mathfrak{N}(\xi^n)}, \tag{3.18a}$$

$$\rho = d_k^{\mathfrak{N}(\xi^n)} + a_k^{\mathfrak{N}(\xi^n)}, \tag{3.18b}$$

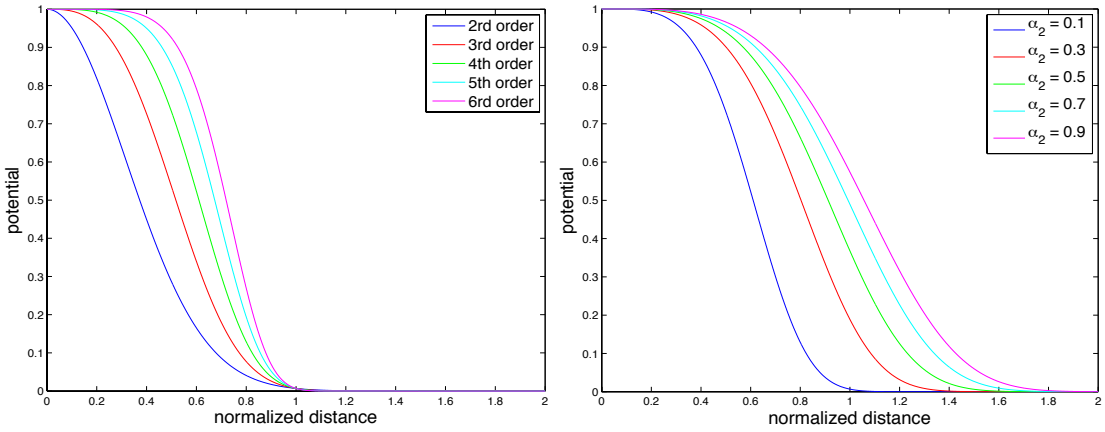
with $d_k^{\mathfrak{N}(\xi^n)}$ as the number of different IDs and $a_k^{\mathfrak{N}(\xi^n)}$ as the number of same IDs with respect to ξ^n within the neighbourhood $\mathfrak{N}(\xi^n)$. δ_k^n is the Dirac function that indicates if n is a new-born particle or not,

$$\delta_k^n = \begin{cases} 1 & if\ \xi^n = 0 \\ 0 & if\ \xi^n \neq 0 \end{cases}. \tag{3.19}$$

In fact, if n is a new-born particle, then $p(L_{\xi^n}) = 1$ with null ID. The ID will be assigned to the new-born particles at the multi-target management stage (Fig. 3.2). The potential $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ is defined as

$$V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n}) = \exp\left\{\frac{-(1-\delta_k^n)(\varphi_k^n)^4}{2\alpha_2}\right\}, \tag{3.20}$$

where the rise $(\cdot)^4$ and α_2 are used to regulate the decreasing trend of the function. The higher α_2, the higher the probability of having an ID far from the group of particles with the same ID.



(a) (b)

Figure 3.5: Potential $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ used for evaluating the distance between particle n and the centre of mass of particles with the same ID. (a) Decreasing trend of the function when changing the order of the exponent of φ_k^n. (b) Changes at different distances from the centre of mass of the particles with the same ID.

φ_k^n is the normalised Euclidean distance from the centre of mass and δ_k^n is defined as in Eq. 3.19.

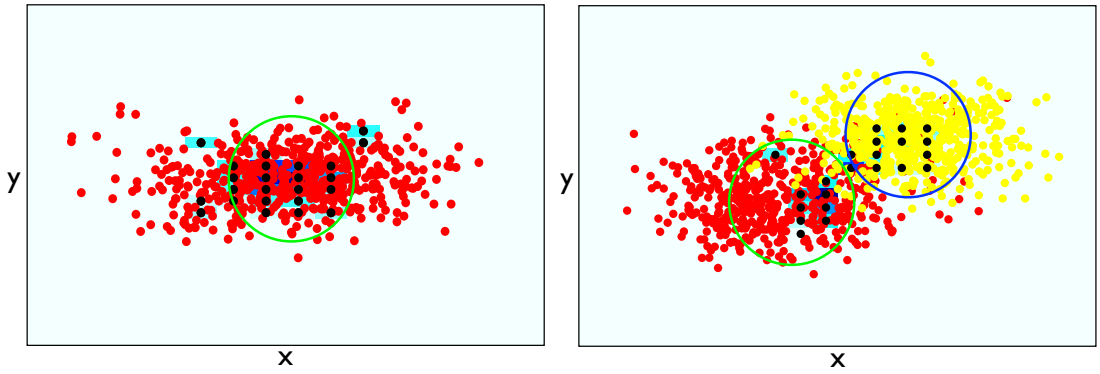
Figure 3.5 shows the trend of the function with different parameters: the horizontal axis represents the variation of φ_k^n and the vertical axis represents $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ as a function of φ_k^n. Figure 3.5(a) shows the decreasing trend of the potential function when changing the order of φ_k^n, whereas Fig. 3.5(b) shows how the potential $V''_{\mathfrak{N}(\xi^n)}(L_{\xi^n})$ changes at different distances from the centre of mass. The centre of mass is calculated by utilising the geometric mean of the position of the particles with the same ξ^n and the normalisation is calculated by taking into account the area of the pixels affected by the blurring introduced during target localisation,

$$\varphi_k^n = \frac{1}{4\Sigma}\sqrt{\left(x_k^n - \sqrt[M]{\Pi_{m=1}^{M} x_k^m}\right)^2 + \left(y_k^n - \sqrt[M]{\Pi_{m=1}^{M} y_k^m}\right)^2}, \tag{3.21}$$

where the normalising factor 4Σ takes into account the 95% of the area affected by the blurring and $M = |\mathfrak{N}(\xi^n)|$ is the number of neighbours of the n^{th} particle. Finally, the value D in Eq. 3.14 used to normalise the energy function for each particle is defined as

$$D(L_{\xi^n}) = \exp\{\alpha_1(1-\delta_k^n)\}. \tag{3.22}$$

The computation of the probability of ξ^n leads to the ID assignment to the new-born particles. The general concept is to assign the same ID as the existing particles within a cluster to the new-born particles that join a cluster. This assignment is based on the probability of existing IDs.



(a) (b)

Figure 3.6: Two sample cases of ID assignment to the new-born particles (black) using Mean-Shift clustering. (a) Cluster (green) containing existing particles with the same ID (red) that is assigned to all the new-born particles within the cluster. (b) Cluster (green) containing existing particles with different IDs (red and yellow) that are assigned to the new-born particles within this cluster using a GMM approach (see text for details).

When existing particles are already initialised with an ID in a cluster, the ID assignment is performed by considering two cases: (i) clusters with existing particles and the same ID (Fig. 3.6(a)), and (ii) clusters with existing particles and different IDs (Fig. 3.6(b)). In the former case, when a cluster contains new-born particles and existing particles sharing the same ID, the ID assigned to the new-born particles is the same as that of the existing particles. In the latter case, when there are new-born particles and existing particles with different IDs in a cluster, we use a method of ID assignment based on GMM[2]. By fitting a GMM with mean components placed on the centre of mass of each group of particles with same ID and variance proportional to the probability of the respective ID, we ensure a *fair* assignment of IDs to the new-born particles located in the cluster. As shown in Fig. 3.4, the widest GMM component belongs to the target with the widest spread function. Likewise, the narrowest GMM component belongs to the target with narrowest spread function. Each fitted Gaussian approximates the spatial distribution of particles sharing the same IDs, and the assignment of the ID to each new-born particle within the cluster is performed according to the Maximum A Posteriori (MAP).

Let us define the set $\mathcal{X}_r = \{(x_k^n, y_k^n, \xi^n) : x_k^n, y_k^n \in \psi_r\}$ of particle locations and IDs belonging to the r^{th} cluster at frame k. Using $\mathcal{X}_r$, we calculate the mean position of the respective IDs, $\theta_\xi \; \forall \; \xi^n \in \mathcal{X}_r$. Let us denote the set of mean positions as $\Theta_r = \{\theta_\xi : \xi^n \in \mathcal{X}_r\}$. We then define the

[2]We choose a probabilistic model, rather than an ad-hoc assignment, since it can be easily extended or replaced with other probabilistic models in the case of different applications of the tracker.

covariance matrices using the total probability of each ID $p(L_{\xi^n})$, such that

$$\phi_\xi = p(L_{\xi^n}) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{3.23}$$

and, as for the mean positions, we define the set of covariance matrices $\Phi_r = \{\phi_\xi : \xi^n \in \mathcal{X}_r\}$. In this way, the fitting is performed using Gaussians with covariances proportional to the probability of the IDs within $\mathcal{X}_r$. Note that $|\Theta_r| = |\Phi_r|$, where $|\cdot|$ is the cardinality of a set. The GMM is defined as a weighted sum of Gaussian densities given by

$$f_{GMM}(\mathcal{X}_r) = \sum_{m=1}^{|\Theta_r|} \pi_m \mathcal{N}(\mathcal{X}_r | \Theta_{r,m}, \Phi_{r,m}), \tag{3.24}$$

where $\Theta_{r,m}$ and $\Phi_{r,m}$ denote the m^{th} mean and covariance component of the corresponding sets, respectively, and each ID $\xi \in \mathcal{X}_r$ is associated with each component m, i.e. $\xi \rightarrow m$. The best fitting of the mixture is performed through the Expectation-Maximization algorithm [67]. Figure 3.4(b) shows an example of GMM fitting when two nearby targets are present. Once the GMM is fitted to the particle locations, the affiliation of the new-born particles to the targets is derived through the calculation of the MAP and the ID assignment is performed with respect to such information. Hence, $\forall\ (x_k^n, y_k^n, \xi^n) \in \mathcal{X}_r$ with $\xi^n = 0$, the ID is assigned using the MAP

$$\xi^n = \bar{\xi} \quad : \ \bar{\xi} \rightarrow m',\ m' = \underset{m=1,\ldots,|\Theta_r|}{\arg\max}\ \{p(m|(x_k^n, y_k^n))\}, \tag{3.25}$$

where $\bar{\xi}$ is the ID associated with the component with the highest probability and

$$p(m|(x_k^n, y_k^n)) = \frac{p(m)p((x_k^n, y_k^n)|m)}{p((x_k^n, y_k^n))} = \frac{\pi_m \mathcal{N}((x_k^n, y_k^n)|\Theta_{r,m}, \Phi_{r,m})}{\sum_{m=1}^{|\Theta_r|} \pi_m \mathcal{N}((x_k^n, y_k^n)|\Theta_{r,m}, \Phi_{r,m})}. \tag{3.26}$$

The state estimate $\hat{\mathbf{x}}_{k|k} = (\hat{x}_{k|k}, \hat{y}_{k|k})$ computed at k given the update is calculated using the weighted sum of the particle positions on their relative weights,

$$\hat{\mathbf{x}}_{\xi,k|k} = \frac{\sum_n w_{\xi,k}^n \cdot [x_{\xi,k}^n\ y_{\xi,k}^n]^T}{\sum_n w_{\xi,k}^n}, \tag{3.27}$$

where the subscript ξ is used to indicate that the state estimate is calculated among particles sharing the same ID. The calculation of $\hat{\mathbf{x}}_{\xi,k|k}$ is used to build the state of each track, where $\forall \xi \in \Xi_k$, we generate the state

$$\hat{\mathbf{x}}_{l,k} = [x_{l,k}\ \ y_{l,k}\ \ \xi_l]^T, \tag{3.28}$$

where the subscript l indexes the ID within the set Ξ_k. Note that the information about the shape is not used by the particle filter. It will be introduced later in Sec. 3.4.

Once the IDs are assigned, the resampling of the particle weights is performed for each cluster independently by assigning the same number of particles to each cluster. Ideally, each cluster contains a single target, hence by resampling each cluster independently we ensure that all clusters/targets evolve over time with the same number of particles.

3.3 Example of likelihood modelling

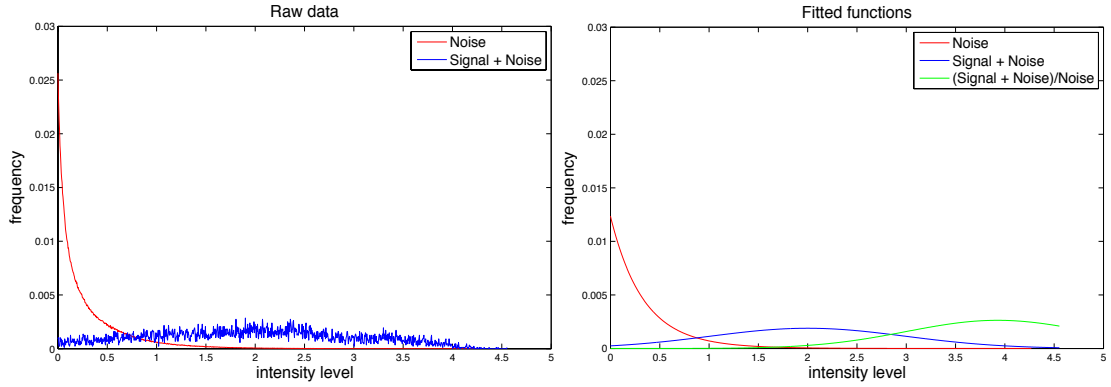
The likelihood function (Eq. 3.12) for MT-TBD is modelled using automatically generated confidence maps filtered by ground-truth information (Sec. 3.2.3). The confidence distribution of true locations is referred to as signal-plus-noise, since manifold factors may affect the response of the target localisation method, such as objects with similar shape or colour. The confidence distribution of false locations is referred to as noise. Ideally, a specific likelihood function should be modelled for each scenario. However, in order to demonstrate the flexibility of the proposed MT-TBD in different scenarios and for different targets, a single likelihood function is defined and used throughout our experiments. In particular, we model the likelihood function using highly noisy data, such as head locations obtained by Support Vector Machine (SVM) [49] and by using HOG features [44] in the TRECVID dataset. The distribution of head/non-head confidences is shown in Fig. 3.7(a). Figure 3.7(b) shows the fitted curves on the data for modelling the likelihood function. The signal-plus-noise distribution is fitted by a Normal distribution and the noise distribution by a Pareto distribution [116]. Since the exponential function goes quicker to zero than the Pareto function, the Pareto distribution is more suited for modelling the noise in Eq. 3.12 (at the denominator). In fact, very high values of likelihood for high values of observed intensities would lead to a divergence in the estimation of the posterior distribution (Eq. 3.11).

The final likelihood ratio (Eq. 3.12) is calculated as

$$\frac{p_{S+N}(\mathbf{z}_k^{(i,j)}|\mathbf{x}_k^n)}{p_N(\mathbf{z}_k^{(i,j)})} = \frac{\sigma_2}{\sqrt{2\pi}\sigma_1}\left(1+\varsigma\frac{\mathbf{z}_k^{(i,j)}}{\sigma_2}\right)^{(1+\frac{1}{\varsigma})}\exp\left\{-\frac{(\mathbf{z}_k^{(i,j)}-h_k^{(i,j)}(\mathbf{x}_k))^2}{2\sigma_1^2}\right\}, \tag{3.29}$$

where σ_1 is the standard deviation of the Normal distribution, and σ_2 and ς are the scale and tail parameters of the Pareto distribution, respectively.

Figure 3.8 shows the effect of the parameter variations on the numerator and denominator



(a) (b)

Figure 3.7: (a) Distribution of the signal-plus-noise (blue) and noise (red) extracted from real data represented by the head locations [44] on the TRECVID dataset. (b) Normal distribution fitted on signal-plus-noise (blue), Pareto distribution on noise (red) and ratio between fitted signal-plus-noise and noise (green).

of Eq. 3.29. When $p_N(\mathbf{z}_k^{(i,j)})$ quickly decreases to zero, i.e. small σ_2 and small ς, the likelihood ratio gives high values. Conversely, when $p_N(\mathbf{z}_k^{(i,j)})$ slowly decreases to zero, i.e. if σ_2 and ς are large, the likelihood gets more biased on the value of the numerator.

3.4 Data-driven postprocessing

We use a shifting temporal window of length τ_w frames that overlaps of one frame over time. The tracks within this temporal window are collected into the set

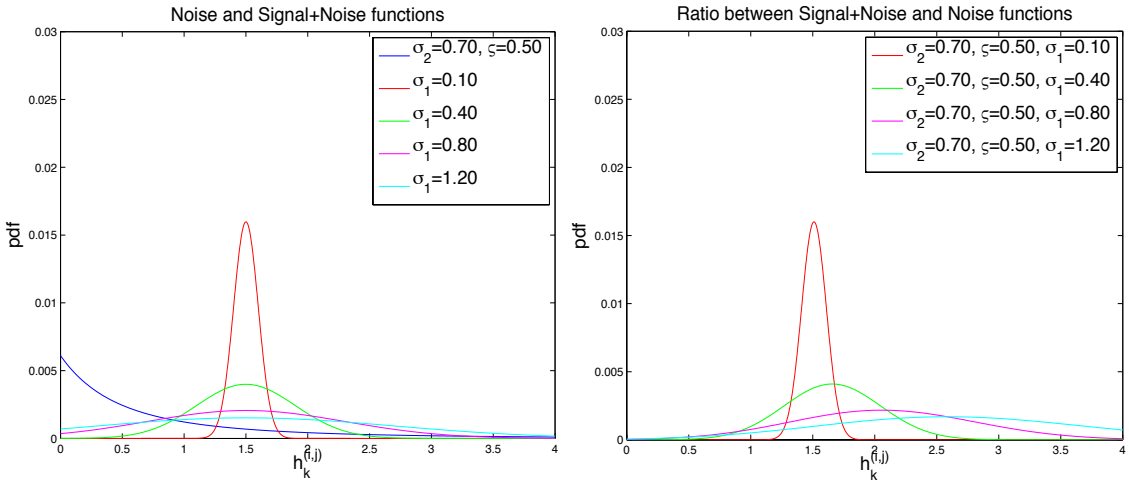
$$\mathfrak{T}_k^{\tau_w} = \{\mathfrak{t}_{l,\mathfrak{K}}^{\tau_w} : \mathfrak{K}_l^{\tau_w} = [k_s, k_e], \mathfrak{K} \subseteq [k-\tau_w+1, k]\}, \tag{3.30}$$

where $\mathfrak{t}_{l,\mathfrak{K}}^{\tau_w}$ is the generic track with ID ξ_l within the interval $\mathfrak{K}_l^{\tau_w} = [k_s, k_e]$ and k_s, k_e are the starting and ending instants of the track within the temporal window, respectively.

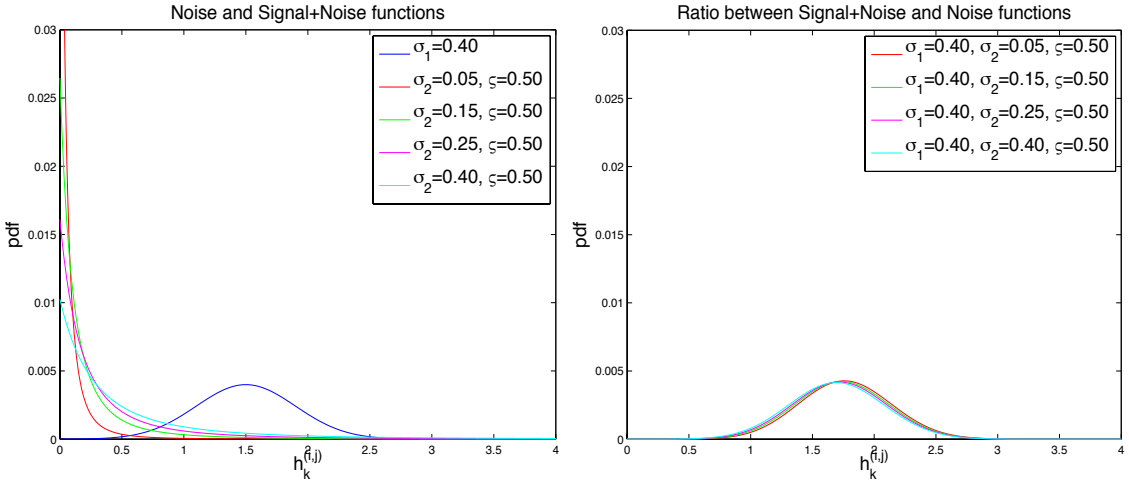
Since we apply postprocessing to an example with extended targets (i.e. people), we include the shape information into the track state. Let the track be defined as

$$\mathfrak{t}_{l,\mathfrak{K}}^{\tau_w} = \{[x_{l,k} \;\; y_{l,k} \;\; S_{l,k} \;\; \xi_l]^T : k \in \mathfrak{K}_l^{\tau_w}\}, \tag{3.31}$$

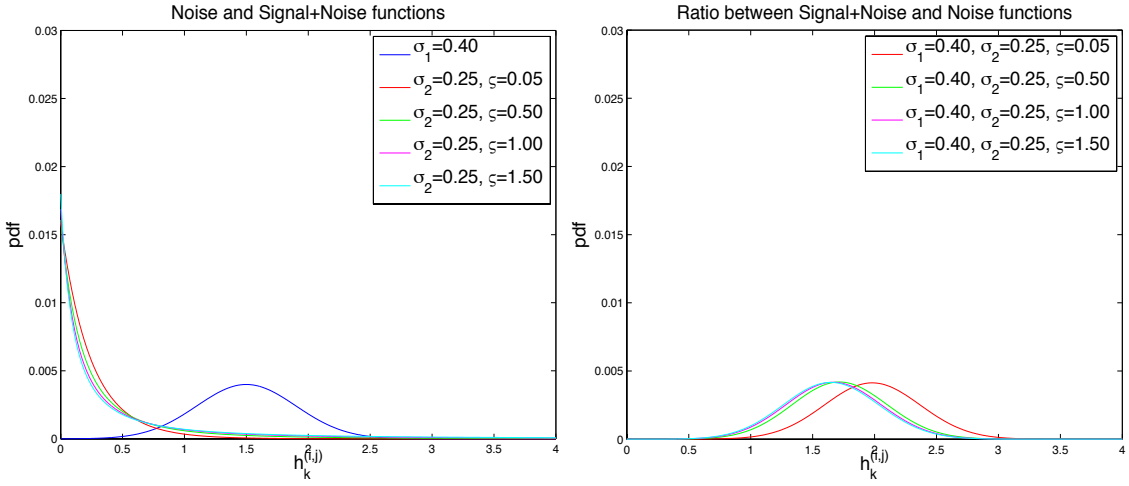
where $(x_{l,k}, y_{l,k})$ corresponds to the top-left corner of the bounding box and $S_{l,k}$ is the bounding box estimated using the scene calibration information. Note that the postprocessing introduces a delay in the tracking output that is analysed in detail in Sec. 3.5.4.



(a)



(b)



(c)

Figure 3.8: Variation of the parameters of the fitted distributions (left) along with the ratios for the likelihood function (right) (Eq. 3.29): (a) σ_1, (b) σ_2 and (c) ς.

The postprocessing stage for multi-person tracking is divided into (i) track pruning to remove tracks with a score s less than 3 within a temporal window $\tau_w^1 = 25$ frames, (ii) track fusion within a temporal window $\tau_w^2 = \tau_w$ and directly proportional to τ_w^1, and (iii) track pruning to remove fused tracks with score less than $\tau_w^2/10$ for a temporal window of τ_w^2.

For track pruning, let us consider a generic track $\mathfrak{t}_{l,\mathfrak{K}}^{\tau_w^1}$ with generic ID ξ_l that exists within a temporal interval of τ_w^1 frames. A score $s_l^{\tau_w^1}$ is assigned to each l^{th} track, such that

$$s_{l,\mathfrak{K}}^{\tau_w^1} = \sum_{k \in \mathfrak{K}_l^{\tau_w^1}} r(\mathfrak{t}_{l,k}^{\tau_w^1}), \tag{3.32}$$

where $r : \mathbb{R}^m \rightarrow \{0,1\}$ and m is a set of rules used to define the score. This leads to $s_{l,k}^{\tau_w^1}$ being equal to the duration of a track (in frames) if $r(\mathfrak{t}_{l,k}^{\tau_w^1}) = 1 \; \forall \, k \in \mathfrak{K}_l^{\tau_w^1}$, otherwise, if $r(\mathfrak{t}_{l,k}^{\tau_w^1}) = 0$ for some $k \in \mathfrak{K}_l^{\tau_w^1}$, the score $s_{l,k}^{\tau_w^1}$ is smaller than the duration of the track. The same process is performed in the temporal window τ_w^2.

In the case of moving cameras, the function $r(\cdot)$ only evaluates the duration of the track in frames, whereas in the case of static cameras, the function $r(\cdot)$ is modelled as a *logic AND* of two rules, $r_1(\cdot)$ and $r_2(\cdot)$, obtained from a background subtraction step. Given $\mathcal{B}(\mathfrak{t}_{l,k}^{\tau_w^1})$, a patch within each bounding box from the difference image between the current frame v_k and the background, we define

$$r_1(\mathfrak{t}_{l,k}^{\tau_w^1}) = \begin{cases} 0 & \text{if } \mu(\mathcal{B}(\mathfrak{t}_{l,k}^{\tau_w^1})) < T_1 \\ 1 & otherwise \end{cases}, \tag{3.33}$$

where $\mu(\cdot)$ calculates the mean pixel intensity and $T_1 = 20$ or $T_1 = 25$ depending on the contrast between targets and background, and

$$r_2(\mathfrak{t}_{l,k}^{\tau_w^1}) = \begin{cases} 0 & \text{if } \sigma(\mathcal{B}(\mathfrak{t}_{l,k}^{\tau_w^1})) < T_2 \\ 1 & otherwise \end{cases}, \tag{3.34}$$

where $\sigma(\cdot)$ calculates the standard deviation of the pixel intensities in grey level and $T_2 = 5$ to remove false positive tracks on flat surfaces such as walls. For the specific case of head tracking, we define an additional rule, $r_3(\cdot)$, to calculate the relative distance and size between bounding boxes in order to remove false tracks originated due to shoulders, when they are erroneously detected as heads.

We formulate the track fusion process as a re-identification problem. The last available position of a track, the velocity and the colour extracted from the upper-body patch [117] are used to find the best match between the final position of a track and the initial position of another track ahead of time.

Let us define a function $\kappa(\cdot)$ that calculates the cost between each track pair within the temporal window τ_2. $\kappa(\mathrm{t}_{l,\Re}^{\tau_w^2}, \mathrm{t}_{l',\Re}^{\tau_w^2})$ is the affinity between track ξ_l and track $\xi_{l'}$, $\forall\, \xi_{l'} \in \Xi_k \setminus \xi_l$. Using the temporal gap between two tracks and the last available position of $\mathrm{t}_{l,\Re}^{\tau_w^2}$, we predict the target position with a linear motion model. The affinity is thus calculated from the end point of a track ($\mathrm{t}_{l,k_e}^{\tau_w^2}$) to the start point of another track ($\mathrm{t}_{l',k_s}^{\tau_w^2}$), with $k_s > k_e$. The calculation of the affinities is based on the Euclidean distance between predicted and current starting point, and the Bhattacharyya distance of the image patch at k_e and that at k_s. The Hungarian algorithm [122] is then iteratively computed to link all the possible track pairs and the set of new tracks is defined as

$$\mathcal{T} = \{\mathsf{T}_a\}_{a=1}^{A}, \tag{3.35}$$

where each T_a is generated by the associated track pairs within $\mathfrak{T}_k^{\tau_w}$ throughout the sequence.

3.5 Analysis of the tracker

In this section, the proposed MT-TBD is tested as multi-person tracker on confidence maps generated by four person localisation algorithms (see Dallar *et al.* [47] for a complete survey on person localisation). In particular, we retrieve person locations using information of: head [20, 44], full-body based on parts [51] and full-body from multiple views [50]. We firstly use reliable confidence maps obtained (i) from head locations guided by the ground truth and (ii) from multiple views of the same scene. Then, we comparatively assess the proposed method with state-of-the-art approaches by employing automatically generated confidence maps on single-view.

3.5.1 Datasets

The experiments are performed on three surveillance videos and two sport videos (Fig. 3.9). The first set of reliable confidence maps are extracted from 2400 frames of size 720×576 pixels from Camera 1 of the London Gatwick airport dataset that is recorded at 25Hz [2] (Fig. 3.9a). The confidence maps are generated as the output of a SVM trained with HOG features [44], where false positive confidences are removed using ground-truth information. Let us call this dataset

Figure 3.9: Sample frames of the datasets used in the experiments: (a) TRECVID, (b) APIDIS, (c) TownCentre and (d) iLids Easy.

TRECVID-HOG+GT. In addition to this, we perform tracking on two different cameras of a basketball scenario (APIDIS dataset [4]) composed of 800 frames of size 800×600 pixels and recorded at 25Hz (Fig. 3.9b). Let us call them APIDISC1 and APIDISC2. Here, the reliable sets of confidence maps are obtained by a multi-layered homography method [50] that exploits the seven cameras available in the dataset. Results on TRECVID-HOG+GT, APIDISC1 and APIDISC2 are reported in Sec. 3.5.3. The results are analysed using MOTA, MOTP, Precision and Recall (Sec. 2.5.4).

MT-TBD is then tested on automatically generated confidence maps on single views (Sec. 3.5.4). Firstly, we use the TownCentre dataset [6] composed of 4500 frames of size 1980×1080 pixels at 25Hz (Fig. 3.9c). For a fair comparison with Benfold and Reid [20], we use the head locations provided by the authors, which are generated using HOG features and SVM. The ground truth has 231 head/person-tracks with an average of 16 people per frame. As the provided person locations have already been thresholded, they are not in the form of confidence values. For this reason, the input to MT-TBD is a confidence map with 2D Dirac delta in correspondence to each localised head. Moreover, we use the iLids Easy dataset [5] composed of 5220 frames of size

Table 3.1: Summary of the datasets and person localisation methods used for validation. Key: H: Head; B: Body; P-B: part-based.

Dataset	Image size	Localisation method	Body part
TRECVID-HOG+GT	720×576	HOG + SVM [44] + GT	H
APIDIS	800×600	Multi-layer homography [50]	B
TownCentre	1920×1080	Binary (HOG + SVM) [44]	H
iLids Easy	720×576	HOG + SVM [51]	B, P-B
TRECVID	720×576	HOG + SVM [44]	H

720×576 pixels at 25Hz, where we obtain person locations using an approach based on body-parts proposed by Felzenszwalb *et al.* [51] (Fig. 3.9d). The ground truth has 17 person-tracks with an average of 1.9 people per frame. Another localisation method based on HOG features and SVM [44] is trained on head patches of 24×24 pixels, and applied to the London Gatwick airport dataset that has the same specifications as above. Let us call this dataset TRECVID to distinguish it from TRECVID-HOG+GT.

Table 3.1 summarises the datasets and the localisation methods used for testing.

3.5.2 Parameters

This section describes the parameters used for MT-TBD. Similarly to Breitenstein *et al.* [27], some parameters are set experimentally.

The choice of the maximum values of velocity, υ_{max}, used to propagate the particles by the proposal density $q_k(\cdot)$ (Sec. 3.2.3) depends on the frame resolution. Higher resolutions lead to higher values of the maximum velocity. TRECVID and iLids Easy datasets have the same frame resolution and, because they contain walking people, the variance of motion is low. For this reason, we set $q_1 \approx 0.3$ and $\upsilon_{max} \approx 3$. Similarly, the TownCentre dataset contains walking people, but the frame resolution is much higher (Tab. 3.1), thus leading to larger displacements on the image plane. Hence, we set $q_1 = 4$ and $\upsilon_{max} = 12$. The noise q_2 associated to the confidence value of the confidence map is then chosen according to the specific confidence map given as input to MT-TBD. The confidence maps of TRECVID, iLids Easy and APIDIS datasets are not thresholded, and we set $I_{min} = 1$, $I_{max} = 3$ and $q_2 \approx 0.3$ for all of them. In the case of TownCentre, the confidence maps are thresholded (there is no variation of confidence) and we set $I_{min} = I_{max} = 2$ with noise $q_2 = 10^{-5}$. The amount of blurring introduced in the target localisation process is modelled by Σ in Eq. 3.5: its value is dependent on the precision of the person localisation method and on the resolution of the confidence map where higher resolution leads to a higher

Table 3.2: Parameters of the likelihood function (Eq. 3.29) used in the experiments.

Dataset	σ_1	σ_2	ς
TRECVID-HOG+GT	0.70	0.10	0.60
TRECVID	0.60	0.30	0.15
iLids Easy	0.15	0.40	1.70
APIDIS	0.70	0.16	0.25
TownCentre	0.80	0.20	0.04

spread in confidence values. For example, $\Sigma = 1.3$ for both TRECVID and iLids Easy datasets that have the same person localisation method and the same frame resolution. On the other hand, in the case of the 2D Dirac delta confidence maps where blurring is absent, $\Sigma = 4$ in order to have a similar spread of the particles over space. The values of α_1 and α_2 for the MRF modelling (Sec. 3.2.4) depend on the desired strength level for maintaining the particles alive in the case of mixing with different IDs. We use $\alpha_1 = 0.2$ and $\alpha_2 = 0.02$ for all the datasets.

The value of σ_1, σ_2 and ς of Eq. 3.29 are provided in Tab. 3.2. The values of σ_1 used in TRECVID-HOG+GT and TRECVID datasets are similar because the same person localisation method is used in both datasets, while the variation of σ_2 and ς is due to the employment of the ground-truth information in TRECVID-HOG+GT. Since in TRECVID-HOG+GT, the noise due to false localisations is absent, we set σ_2 and ς such that the numerator (signal-plus-noise) of the likelihood function is predominant on the denominator (noise). Conversely, in the case of TRECVID, the confidence maps are more noisy, and σ_2 and ς are set in order to take into account also the contribution of the denominator. The person localisation method used in iLids Easy [51] provides a more stable signal-plus-noise compared to the method used in TRECVID, thus leading to a smaller variance of the confidence values and hence to a smaller σ_1. However, the noise is still high and σ_2 is set as for TRECVID. The value of ς is large, in order to avoid the divergence of the likelihood function in the case of large confidence values. For APIDIS and TownCentre the confidence maps are provided as 2D Dirac delta functions and this justifies the similarity of σ_1 and σ_2 values. The parameters are chosen such that the likelihood function does not diverge. Unlike TownCentre where the 2D Dirac deltas are binary, in APIDIS the 2D Dirac deltas represent confidence values and, similarly to the iLids Easy, we keep the value of ς large in order to avoid the divergence of the likelihood function for large confidence values.

3.5.3 Analysis of the steps

The validation of the proposed method is performed with the tracking results generated by (i) MT-TBD without any postprocessing, (ii) track pruning on the tracks from MT-TBD, (iii) track fusion on the tracks from the previous track pruning, and (iv) track pruning on the tracks from the previous track fusion.

The analysis of the tracking results generated by MT-TBD without any postprocessing shows the behaviour of the proposed filter, especially in situations with close targets where the MRF modelling helps to avoid particles of different targets being mixed together. The first dataset we employ is the TRECVID-HOG+GT. In Fig. 3.10, a situation of a significant overlap ($> 50\%$) between two targets is shown. In Fig. 3.10(a), all targets in the scene are correctly tracked. Subsequently, when two targets get closer (Fig. 3.10(b)), the target further away from the camera gets almost completely occluded, however, since the confidence map still localises the target, MT-TBD correctly tracks it. In Fig. 3.10(c), when the targets are completely overlapped, the confidence values on the confidence map appear as a single target with a large spread. Even if the tendency for mixing of particles with different IDs is visible, the MRF modelling consistently assigns the correct ID to each particle. Figures 3.10(d-f) finally show how the particles remain associated to the correct target over time.

Figure 3.11 shows an example of incorrect ID assignment leading to an ID switch generated by MT-TBD without any postprocessing. In this case, the confidence values are completely overlapped with a mixing of IDs. Initially, two close targets move in the same direction (Fig. 3.11(a)) and suddenly one target changes direction and becomes completely occluded (Fig. 3.11(b)). Although both IDs remain alive for a few time steps, the particles with magenta ID die (Fig. 3.11(d)) and the green particles move on the visible target. When the occluded target becomes visible again on the confidence map (Fig. 3.11(e-f)), MT-TBD immediately initialises a new track and correctly tracks the target in the following frames. Note that MT-TBD is not designed to reinitialise a target track with a previously existing ID, hence a different ID is assigned to a target that disappears and reappears in a scene, thus leading to an ID switch.

Quantitative results for MT-TBD and postprocessing are reported in Fig. 3.12(a). After the first track pruning, Recall and MOTA are slightly decreased because the short tracks are removed due to their low score. However, after track fusion has been applied, Recall reaches a higher value because short but reliable tracks are correctly fused. Lastly, by pruning the unreliable

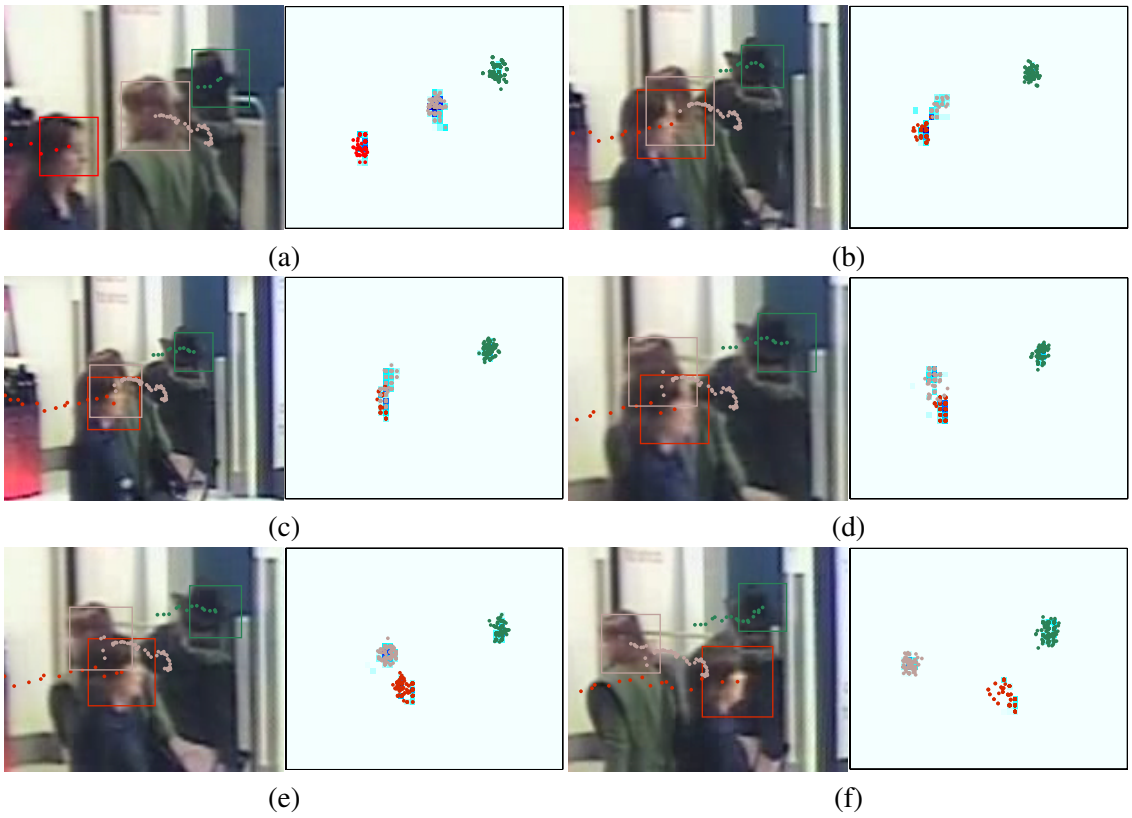


Figure 3.10: Example from TRECVID-HOG+GT dataset which represents a situation of a significant overlap ($> 50\%$) between two targets (red and grey colour-codes). Before the occlusion occurs (a) the targets are correctly tracked with unique IDs. When the occlusion starts (b) particles start mixing but the IDs are still well-separated. During the occlusion (c), particles and IDs are mixed, but it is possible to notice that the mixing remains limited. When the targets start splitting (d), there is a tendency for the particles to mix (the red particles mix with the grey particles). When the split of targets occurs (e-f), the particles are again well-separated with their own IDs.

tracks generated by the fusion stage, it is possible to keep the same value of Recall while increasing Precision. An improvement in terms of ID switches that is due to the linking (fusion) of fragmented trajectories can be seen throughout these steps.

The second validation of MT-TBD and postprocessing is presented using APIDISC1 and APIDISC2 (Fig. 3.12). By analysing the results shown in Fig. 3.13(a-d), we see the tracking succeeding in most cases even while players are very close to each other. The main challenges here are the sudden movements of players. Recall is larger than 90% in both datasets even if some of the tracks are lost (Fig. 3.13(d)). A possible solution for this problem is the use of multi-dynamic model particle filters [142], which are able to perform nonlinear filtering with switching of dynamic models.

We qualitatively showed the behaviour of the proposed particle filter in the case of full and

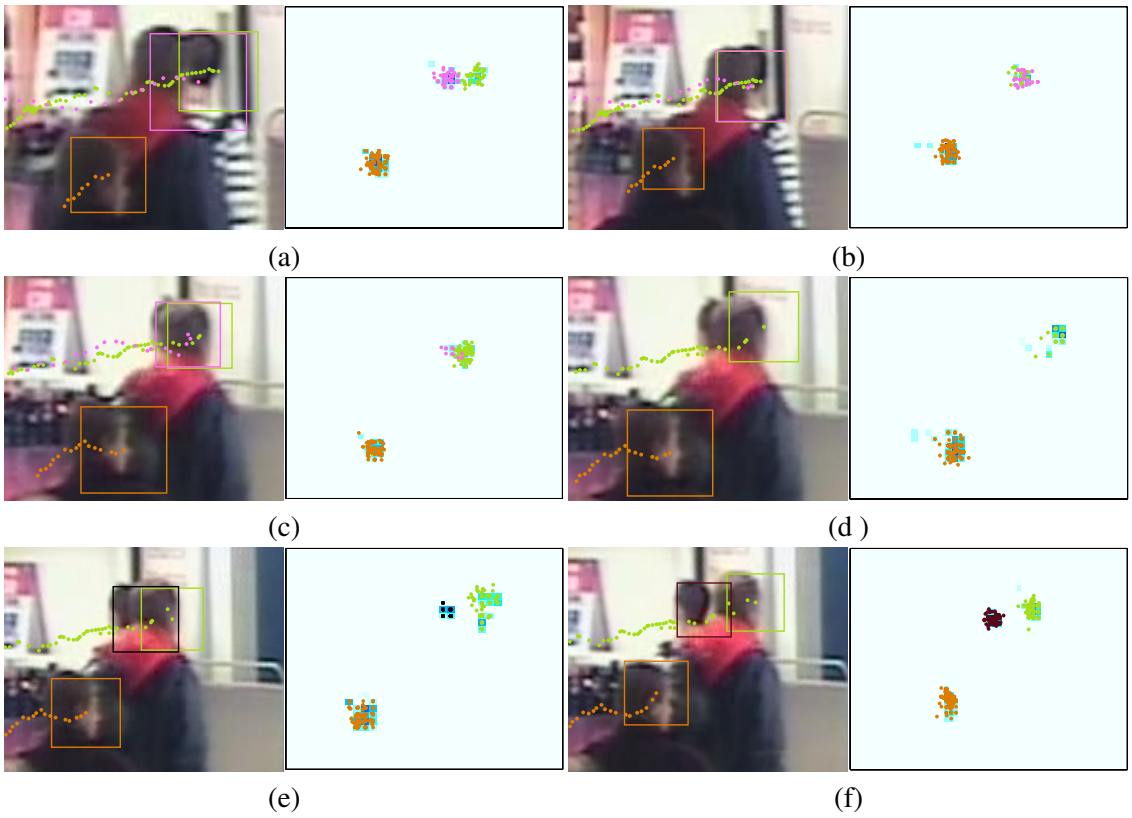


Figure 3.11: Example from TRECVID-HOG+GT dataset which represents a situation of significant overlap ($\approx 100\%$) between two targets where an ID switch occurs. Before the occlusion (a), the targets are correctly tracked with unique IDs. During the occlusion (b), the particles are mixed and the algorithm cannot maintain the correct IDs. When the targets start splitting (c), the number of magenta particles start getting smaller. Then particles belonging to the magenta target die (d) and the green particles swap target (they get attached to the target in front). When the target that is behind becomes visible, MT-TBD immediately starts tracking it again but with a new ID (e-f).

partial mixing of the particles. On the one hand, when the confidence values of two different targets fully overlap (one target gets fully occluded) and the particles mix, it is likely to lose one of the targets. This loss depends on the total probability of the particles assigned to each of the targets, and the target with the lowest ID probability is the most likely to be lost. On the other hand, when the confidence values of the two targets partially overlap and some of the particles mix, it is likely that the tracker is able to maintain the target identities. We then quantitatively showed how postprocessing tailored for a specific application (i.e. people tracking) can improve the results, especially in terms of ID switches (IDS) (Fig. 3.12). From the results, we can infer that most of IDS are due to fragmented tracks and false positives. In fact the fusion and pruning process was effective in reducing the number of IDS.

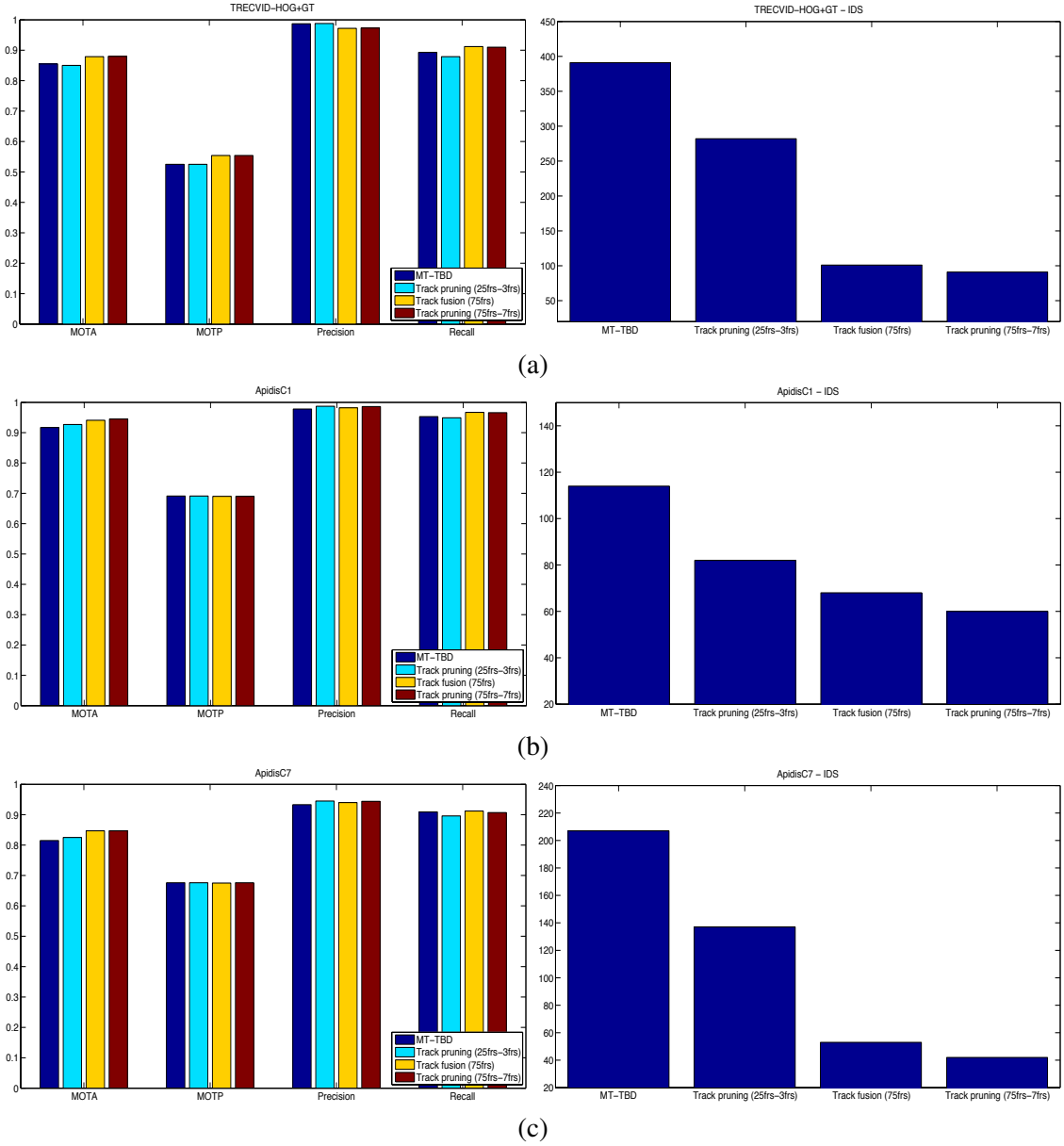


Figure 3.12: Tracking results of the proposed method at different stages of computation: MT-TBD, Track pruning τ_w^1, Track fusion τ_w^2 and Track pruning τ_w^2. Left: evaluation performed in terms of MOTA, MOTP, Precision and Recall. Right: variation of ID switches (IDS). Dataset: (a) TRECVID-HOG+GT, (b) APIDISC1 and (c) APIDISC2. The numbers in brackets refer to the length of the temporal window and to the threshold applied on the minimum track length in the track pruning stage.

3.5.4 Sensitivity analysis

As far as the TownCentre dataset is concerned, we show how our method outperforms the recent work by Benfold and Reid [20] by using the same observations for tracking. We name their method as MCMCDA. TownCentre is fairly challenging as it contains very close targets and the field-of-view of the camera is very large, hence ID switches are likely to be frequent. For compar-

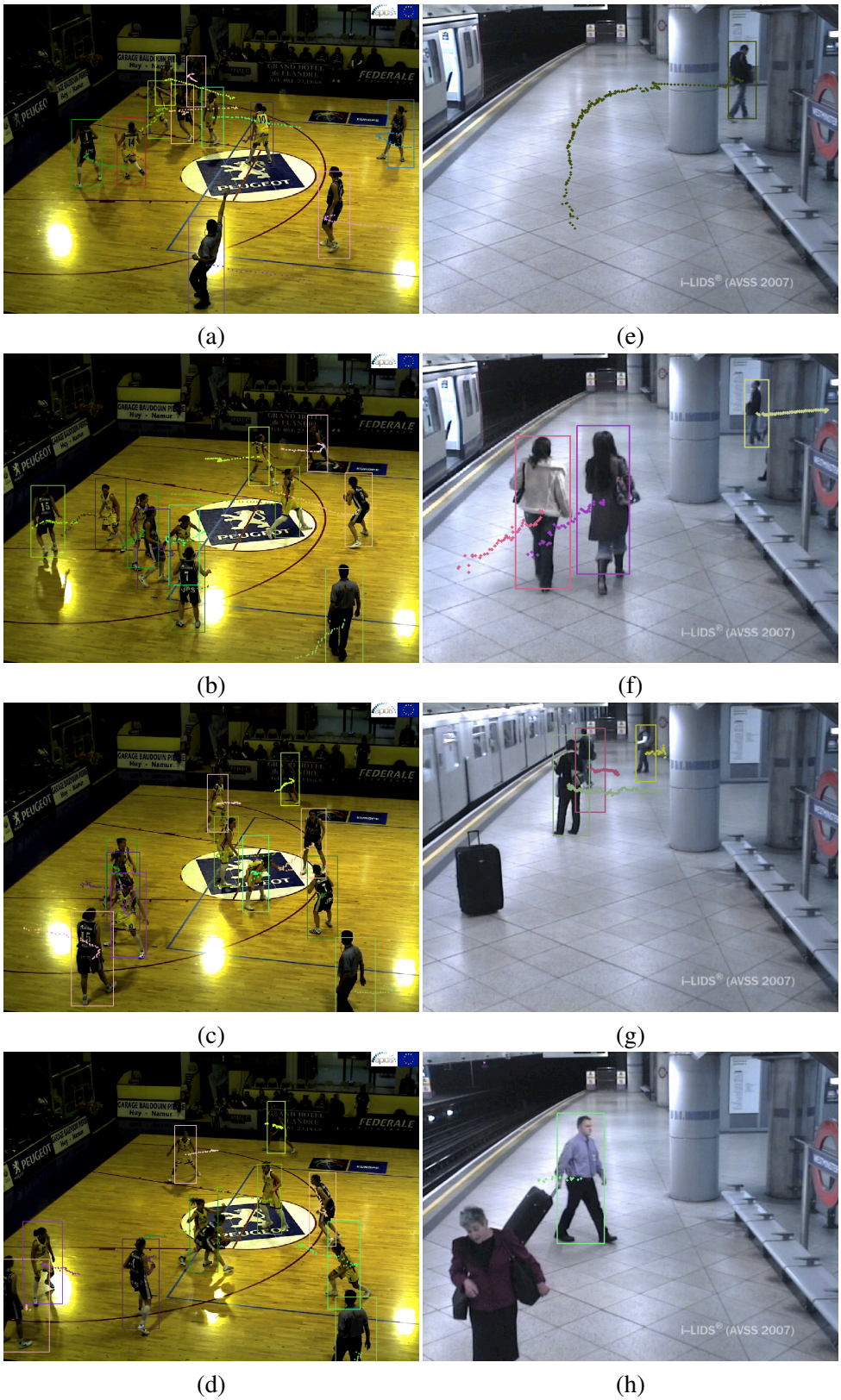


Figure 3.13: Sample tracking results of the proposed method on (a-d) APIDISC2 and (e-h) iLids-Easy datasets. The visualisation of tracks for APIDISC2 are truncated to the last 50 frames to make the examples clearer. The tracks for iLids are shown from the initialisation of the track.

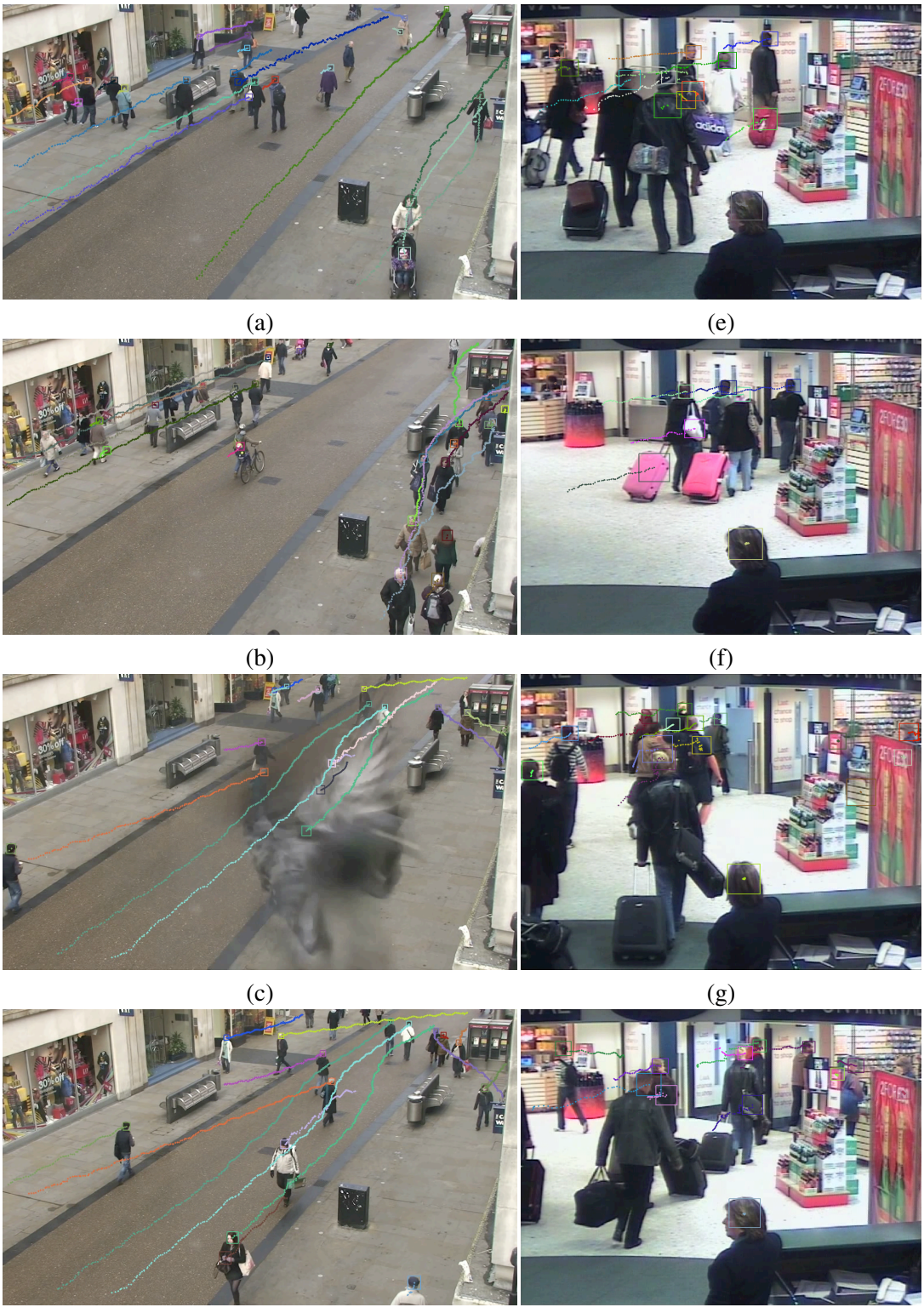


Figure 3.14: Sample tracking results of the proposed method on (a-d) TownCentre and (e-h) TRECVID datasets. The visualisation of tracks for TRECVID are truncated to the last 50 frames to make the examples clearer. The tracks for TownCentre are shown from the initialisation of the track.

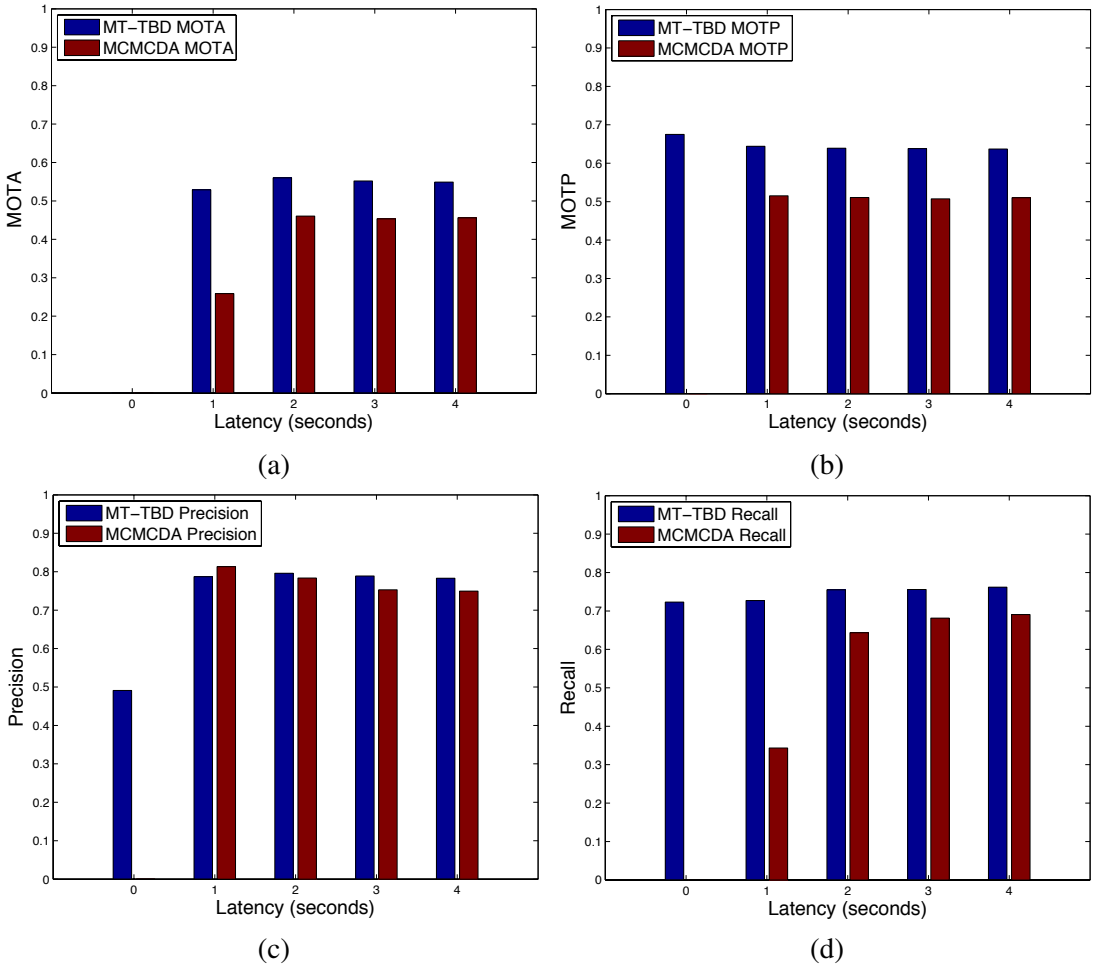


Figure 3.15: Comparison of our results on TownCentre dataset with the Benfold and Reid method [20] (MCMCDA). The graphs show the variation of the scores as a function of the latency introduced by the postprocessing: (a) MOTA, (b) MOTP, (c) Precision and (d) Recall.

ison, we present the results with the same latency used in [20] for postprocessing, specifically 1, 2, 3, and 4 seconds (1 second = 25 frames). In order to show the global improvement of our proposed method, we also include the performance of MT-TBD without any postprocessing. Note that, unlike our tracker, MCMCDA cannot work with no latency. Figure 3.15 shows the quantitative results. The superior performance of the proposed method is highlighted by the value of Recall that is consistently higher than MCMCDA at various latencies. For MT-TBD without latency (and no postprocessing), the value of Recall is already comparable with that of 4-second latency. However, Precision in this case is lower due to the short and false tracks generated by the temporally-consistent false positive head locations. By applying the proposed postprocessing, Precision drastically increases. Figure 3.14(a-d) shows sample tracking results where the method is robust under severe occlusions with few fragmented tracks.

The results of iLids Easy and TRECVID are quantitatively evaluated in Fig. 3.16. For these two cases, the input confidence maps to MT-TBD are given as scalar confidence values. For this reason, it is possible to analyse the results in detail by comparing the accuracy of target localisations with the accuracy of MT-TBD. In the graphs of Fig. 3.16(c)-(d), the variation of Precision and Recall of the localisation results with respect to the threshold variation on the confidence maps is shown, and the improvement that MT-TBD achieves can be appreciated. With the iLids Easy dataset, an indoor video surveillance scenario is analysed where the main challenges are due to (i) the perspective of the scene (which leads to occlusions among targets), (ii) a column in the middle of the scene (which causes complete occlusions), and (iii) a dynamic background (which does not allow an effective background subtraction). Since a full-body person detector [51] is used, half-visible people in the scene cannot be localised, thus leading to the failure of our multi-person tracking in the lower part of the image (Fig. 3.13(h)). The graph in Fig. 3.16(c) shows that the maximum value of Precision is about 0.6 in person localisation and the maximum value of Recall is about 0.8. The MT-TBD, in this case, can achieve Precision of 0.490 and Recall of 0.676, while the postprocessing considerably increases Precision while maintaining high values of Recall. In Fig. 3.13(e-f)) it is possible to see how track fusion allows tracking in the case of complete occlusions.

On the TRECVID dataset, we validate the proposed method using a confidence map built on head localisations. The head localisation reduces the effect of occlusions among targets in crowded scenarios, but since many objects in the scene have shapes similar to heads (e.g. bags, shoulders and luggage), the localisation contains a large number of false positives (Fig. 3.16(d)). In comparison with the iLids dataset (Fig. 3.16), Precision remains higher since the spread function of the localised heads is smaller than the person localisations in iLids. Hence, head localisation turns out to have higher Precision than that for bodies at same Recall values. Qualitative tracking results are shown in Fig. 3.14(e-h): it is possible to notice the long tracks belonging to the heads and the false positive tracks. The quantitative evaluation is given in Fig. 3.16(b,d). The improvement of the tracker with respect to the confidence map is shown in Fig. 3.16(d), where Recall of 0.813 and Precision of 0.324 are achieved. Then, the postprocessing phase improves the Precision rate by around 20% at the cost of a slight decrease in Recall.

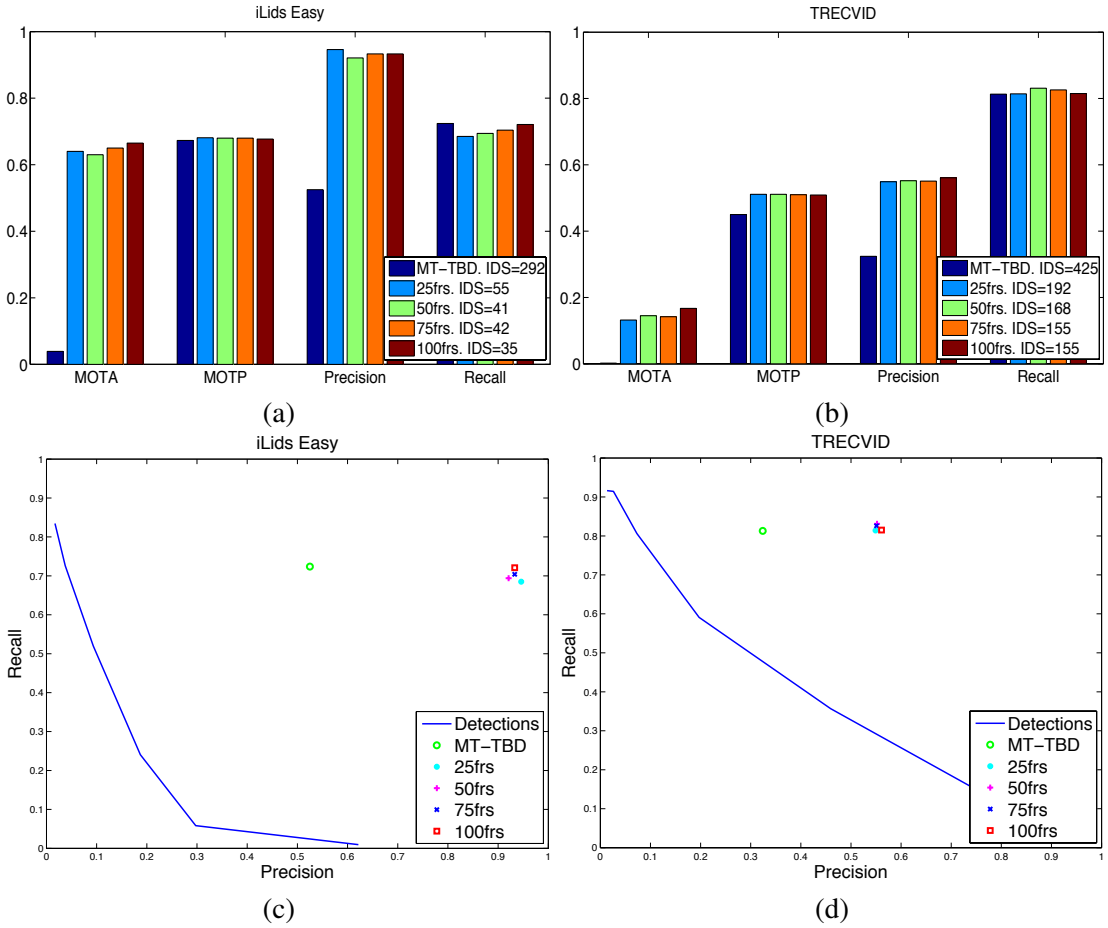


Figure 3.16: Results of the proposed tracker and person localisation methods obtained on iLids Easy and TRECVID datasets. (a-b) Bar plots of MOTA, MOTP, Precision and Recall values by varying the temporal window τ_w used in the postprocessing. (c-d) Precision and Recall rates for the thresholded confidence map plotted along with the tracking scores that show tracking performance with respect to the input with varying threshold. The duration of the temporal window is indicated in frames (frs) within the legend. IDS: ID Switches.

3.5.5 Computational cost

The overall complexity of MT-TBD with N particles has an upper bound of $O(N\log(N))$ operations. Specifically, for the motion model, the proposal distribution and the multinomial resampling (Sec. 3.2.1 and 3.2.3), the cost is $O(N)$ as these operations are sequential on the number of particles. For the neighbourhood search of Eq. 3.15, we give as input a set of spatially ordered particles at the cost of $O(N\log(N))$ and we use a method based on binary search [40] whose cost is $O(\log(N))$. For the Mean-Shift clustering, the operation is performed on the complete set of N particles with complexity $O(N\log(N))$ [56].

As far as online trackers are concerned, the difference between the computational complexity of MT-TBD and that of a multi-particle filter is the $\log(N)$ factor. A particle filter for single target

tracking can run with $O(N)$ operations [34]. Multi-particle filters, where a filter is initialised for each target [27, 159], can then run with complexity $O(MN)$, with M being the number of estimated targets. Therefore MT-TBD becomes more convenient than a multi-particle filter with increasing numbers of targets (i.e. $M > \log(N)$). There may also be methods to perform the detection association between consecutive frames and the complexity may be $O(D^3)$ with the Munkres algorithm [122], where D is the number of measurements. However, this algorithm is unable to deal with noise. Robustness to noise can be achieved using, for example, data association based on Multi Hypothesis Tracker [25, 139] whose complexity is $O((MD)^2)$.

3.6 Summary

In this chapter we presented a Bayesian method for multi-object tracking based on track-before-detect, which utilises the Markov Random Fields applied on the particle states to perform tracking, (i) of unknown and unlimited numbers of targets and (ii) by probabilistically managing the ID assignment with close objects. To deal with close targets, our approach does not rely on appearance information, but performs a probabilistic optimisation to keep the ID associated to the correct targets. The proposed MT-TBD utilises measurements coming from confidence maps and performs tracking of the noisy confidence values. The targets are assumed to be point targets with a noise spread function modelled as a 2D Gaussian distribution. The tracking is performed using a Bayesian recursion via prediction and update steps. The prediction is described with a linear motion model where the unforeseen disturbances are modelled as Gaussian noise. The update is performed through a likelihood function modelled on a particular controlled scenario and used for all our experiments. The birth and death of the particles at each iteration of the filter is modelled with Markov Random Fields, which assumes the Markovian property. The state estimate of a target is performed via Mean-Shift clustering and supported with Mixture of Gaussians in order to allow an accurate assignment of IDs within each single cluster. We assessed the behaviour of the method on surveillance and sport datasets with different perspective views, partial and full occlusions of targets, different backgrounds and variable numbers of people. We evaluated the performance of (i) MT-TBD without any postprocessing, (ii) track pruning on the tracks from MT-TBD, (iii) track fusion on the tracks from the previous track pruning, and (iv) track pruning on the tracks from the previous track fusion.

Experiments showed that the inclusion of the ID inside the particle state was effective in

the case of partially overlapping confidence values, but was not reliable in the case of a full overlap. The postprocessing tailored for people tracking improved the tracking performance, especially by lowering the number of ID switches. We discovered that most of the ID switches were due to fragmented tracks and that the track fusion allowed halving of the number of ID switches. We analysed the sensitivity of the method by varying the buffer size and comparing the performance with an alternative method from the state-of-the-art based on data association. MT-TBD outperformed the other method for different buffer sizes. The comparison of MT-TBD with state-of-the-art multi-target trackers is extensively performed in the next chapter after the presentation of a new set of measures. The analysis is performed in the next chapter since we first want to describe the new measures and then compare the tracking results using the proposed evaluation framework.

Chapter 4

Performance evaluation of multi-target tracking

4.1 Introduction

In the case of extended multi-target tracking, the state-of-the-art lacks of a set of measures that holistically evaluate target-size changes over time and that are parameter independent. These properties are key to achieve an effective and unbiased assessment of the performance. One should be able to quantify the assessment with a single-valued score as well as having the possibility of displaying more detailed results on a range of values. The parameter independency is an important aspect because it enables application-unbiased assessments. Generally, parameter-dependent measures need the definition of a threshold for the overlap between ground-truth and estimated bounding boxes in order to define a correct match. This threshold has to be different according to the type of application for which the tracking results are evaluated. For example, the threshold used to assess head tracking applications is different from that used for body tracking [20]. The evaluation of changes in target size over time has to be quantified in order to infer the accuracy with which a target is tracked. Also, the capability of a tracker to distinguish targets and maintain the tracking locked on the same target has to be quantified.

In particular, the three relevant aspects to be evaluated are accuracy, cardinality and number of ID changes [81, 143]. The *accuracy* quantifies the closeness of agreement between estimated and ground-truth states [8], and it can be calculated as an error score (i.e. distance [143], overlap [24, 81]) or based on true positives (correct estimations), false positives (incorrect estimations) and false negatives (missed estimations) [110]. It is also important to calculate the accuracy at frame

level as well as at sequence level (long-term accuracy) in order to have a broader understanding of the performance. The accuracy error can be calculated by solving the assignment (association) problem (e.g. Hungarian algorithm [92]) between estimated and ground-truth targets. This error cannot be quantified when there is a mismatch in the number of estimated or ground-truth targets. Therefore, the largest accuracy error for a target which is non-associated and either estimated or ground-truth has to be included in. The *cardinality error* indicates the difference between the number of (correctly and incorrectly) estimated and ground-truth targets, and allows the largest accuracy error to be explicitly accounted for. *ID changes* measure the incorrect associations between estimated and ground-truth targets.

In this chapter, we present three novel overlap-based measures for multiple extended-target video trackers that evaluate tracking performance at frame level, accounting for (i) accuracy and cardinality errors; (ii) long-term tracking accuracy using lost-track-ratio information; and (iii) ID changes in a parameter-independent manner[1]. The proposed measures aim to address the limitations discussed in Sec. 2.6. The proposed measures are extensively validated by comparing them with existing measures and in the form of evaluation of MT-TBD with respect to three state-of-the-art multi-target trackers on challenging real-world datasets.

The chapter is organised as follows. Section 4.2 describes the three proposed measures, where the multiple extended-target tracking error measure is presented in Sec. 4.2.1, the multiple extended-target lost-track ratio measure in Sec. 4.2.2 and the normalised ID changes measure in Sec. 4.2.3. In Sec. 4.3, the experimental validation of the measures along with the comparison of the trackers is performed. Section 4.4 summarises the achievements.

4.2 Tracking error measures for extended targets

4.2.1 Multiple extended-target tracking error

The overlap-based Multiple Extended-target Tracking Error (METE) measure combines accuracy and cardinality errors in a parameter-independent manner. The spatial overlap information in METE allows us to discard OSPA parameters (Eq. 2.10), namely the penalty (p) for the estimated states located far away from the ground-truth states and the cut-off parameter (c) that defines the upper bound discrepancy.

[1]The work in this chapter appears in [J1] and was jointly performed with another PhD student who was the first author of the paper. The splitting of the contribution between the first and the second author is 60-40%.

The accuracy error, $\mathcal{A}_k$, represents the extent of the mismatch between estimated and ground-truth states at frame k and is defined as

$$\mathcal{A}_k = \min_{\pi \in \Pi_{max({}^g u_k, u_k)}} \sum_{d=1}^{min({}^g u_k, u_k)} \left(1 - O({}^g S_{d,k}, S_{\pi(d),k})\right), \tag{4.1}$$

where $O({}^g S_{d,k}, S_{\pi(d),k}) = \frac{|{}^g S_{d,k} \cap S_{\pi(d),k}|}{|{}^g S_{d,k} \cup S_{\pi(d)k}|}$ defines the amount of spatial overlap between ${}^g S_{d,k}$ and $S_{\pi(d),k}$; and $O(\cdot) \in [0,1]$ [110], like in Eq. 2.17. Without loss of generality, we consider here ${}^g S_{d,k}$ and $S_{\pi(d),k}$ to be bounding boxes. $\Pi_{max({}^g u_k, u_k)}$ is the set of permutations, each of which contains $min({}^g u_k, u_k)$ elements, drawn from $\{1,2,\ldots,max({}^g u_k, u_k)\}$. The permutation that minimises the summation term in Eq. 4.1 defines the association between estimated and ground-truth states while contributing to the computation of the accuracy error at frame k. This minimisation is performed by the Hungarian algorithm [92]. $\mathcal{A}_k \in [0, u_k = {}^g u_k]$ when $u_k = {}^g u_k$; $\mathcal{A}_k \in [0, {}^g u_k]$ when $u_k > {}^g u_k$ (i.e. the association is performed only for the ${}^g u_k$ terms); and $\mathcal{A}_k \in [0, u_k]$ when $u_k < {}^g u_k$ (i.e. the association is performed only for the u_k terms). Since $\mathcal{A}_k$ does not account for the discrepancy between u_k and ${}^g u_k$ (i.e. unassociated targets) in the case of $u_k > {}^g u_k$ and $u_k < {}^g u_k$, the accuracy error is computed for the associated pairs only. Hence, we calculate the discrepancy between the number of estimated and ground-truth targets, namely the cardinality error $\mathcal{C}_k$:

$$\mathcal{C}_k = |u_k - {}^g u_k|. \tag{4.2}$$

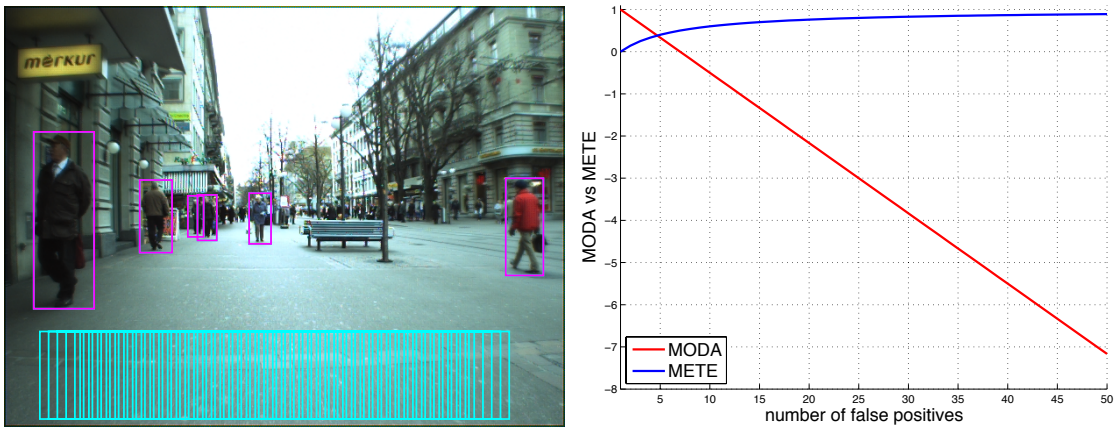
We combine $\mathcal{C}_k$ with $\mathcal{A}_k$ to consider the unassociated targets within the evaluation procedure [143,153] and to provide a single-score performance evaluation at frame level. METE is therefore computed as:

$$\text{METE}_k = \frac{\mathcal{A}_k + \mathcal{C}_k}{max({}^g u_k, u_k)}, \tag{4.3}$$

where $\text{METE}_k \in [0,1]$: the lower METE_k, the better the tracking result. We explain below the bounds of the measure, where $\text{METE}_k = 0$ for the best tracking case and $\text{METE}_k = 1$ for the worst tracking case.

Best tracking case: $\mathcal{A}_k = 0$: $O(\cdot) = 1$ for all the associated pairs (Eq. 4.1), and $\mathcal{C}_k = 0$ since $u_k = {}^g u_k$ (Eq. 4.2). This implies $\text{METE}_k = 0$, using Eq. 4.3.

Worst tracking case: $\mathcal{A}_k$ has its maximum value, i.e. $\mathcal{A}_k = u_k = {}^g u_k$ when $u_k = {}^g u_k$, $\mathcal{A}_k = {}^g u_k$ when $u_k > {}^g u_k$ (the association is performed only for the ${}^g u_k$ terms) and $\mathcal{A}_k = u_k$ when $u_k < {}^g u_k$



(a) (b)

Figure 4.1: Example showing the unbounded behaviour of MODA. (a) Sample frame from ETH Bahnhof [7] (six targets). Ground-truth and tracker's estimates are shown as magenta and cyan bounding boxes, respectively. The estimated bounding boxes are overlapping with corresponding ground-truth bounding boxes. (b) MODA and METE values are calculated while gradually increasing false positives in the lower part of the frame (cyan bounding boxes). MODA decreases without lower bound (Eq. 2.18), whereas METE $\in [0,1]$.

(the association is performed only for the u_k terms). Thus the numerator of Eq. 4.3 becomes $\mathcal{A}_k+\mathcal{C}_k =^g u_k = u_k : u_k = {}^g u_k$ meaning $\mathcal{C}_k = 0$; $\mathcal{A}_k+\mathcal{C}_k = v_k + |u_k - {}^g u_k| = u_k : u_k > {}^g u_k$; $\mathcal{A}_k+\mathcal{C}_k = u_k + |u_k - {}^g u_k| =^g u_k : u_k < {}^g u_k$. Therefore, $\mathcal{A}_k+\mathcal{C}_k = max({}^g u_k, u_k)$, which implies $\text{METE}_k = 1$, using Eq. 4.3. The other tracking cases lie within the two bounds of METE as shown in Fig. 4.1.

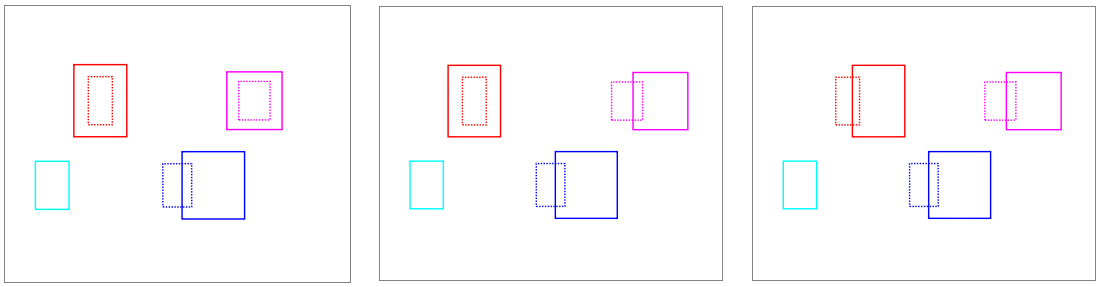
Since same METE values for two trackers may be generated by different combinations of accuracy and cardinality errors, it is useful to analyse these errors separately in order to better understand their individual influence in the calculation of METE. Therefore, we use two error rates, the Accuracy Error Rate (AER):

$$\text{AER} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{A}_k \tag{4.4}$$

and the Cardinality Error Rate (CER):

$$\text{CER} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{C}_k. \tag{4.5}$$

To conclude, METE allows the evaluation of target-size changes, whereas MODA is unable to calculate them. METE is parameter-independent and numerically bounded between 0 and 1,



MODA = -1.33
METE = 0.81
(a)

MODA = -1.33
METE = 0.88
(b)

MODA = -1.33
METE = 0.94
(c)

Figure 4.2: Example of MODA limitations [81]. Although the three cases are distinguishable, MODA is unable to discriminate them [20]. Ground-truth and estimated boxes are shown as dotted and solid lines, respectively. However, the proposed measure METE can distinguish the three cases.

whereas MODA is parameter dependent and not lower bounded (Fig. 4.1). We observed that the parameter dependence of MODA may limit the discrimination of different tracking results (Fig. 4.2).

4.2.2 Multiple extended-target lost-track ratio

The Multiple Extended-target Lost-Track ratio (MELT) evaluates the tracking accuracy across the sequence by using parameter-independency and enabling analysis at different accuracy levels. Given $\mathcal{T}$ and ${}^{g}\mathcal{T}$, the association is performed at frame and by minimising the cost $(1-O(\cdot))$ computed for all pairs of estimated and ground-truth targets. The minimisation process uses the Hungarian algorithm as in Eq. 4.1. The procedure involves a unique assignment at frame level, whereas at track level a ground-truth track may be associated to multiple estimated tracks due to fragmentations and/or ID changes.

The accuracy is calculated at track level by computing the lost-track ratio ($\lambda_d^{\tau_{ltr}}$) [110] for each associated pair of ground-truth track d and estimated track(s) as follows:

$$\lambda_d^{\tau_{ltr}} = \frac{N_d^{\tau_{ltr}}}{N_d}, \tag{4.6}$$

where $N_d^{\tau_{ltr}}$ is the number of frames with spatial overlap $O(\cdot) \leq \tau_{ltr} : \tau_{ltr} \in \mathbb{R}_{(0,1]}$ between the associated pair and N_d is the total number of frames in the ground-truth track d. $\lambda_d^{\tau_{ltr}} \in [0,1]$; the lower $\lambda_d^{\tau_{ltr}}$, the better the performance. We compute the lost-track ratio for a range of a finite number of τ_{ltr} values and obtain $\lambda_d(\tau_{ltr}) = \{\lambda_d^{\tau_{ltr}}\}_{\tau_{ltr} \in \mathbb{R}_{(0,1]}}$ such that the total number of

sampled τ_{ltr} values is $\Upsilon_{\tau_{ltr}}$ (required for numerical approximation). We compute $\lambda_d(\tau_{ltr})$ for all ${}^{g}U$ ground-truth tracks to generate the matrix Λ:

$$\Lambda = [\lambda_d^{\tau_{ltr}}]_{{}^{g}U \times \Upsilon_{\tau_{ltr}}}, \tag{4.7}$$

where ${}^{g}U$ and $\Upsilon_{\tau_{ltr}}$ are the number of rows and columns of the matrix, respectively. We quantify tracking performance by defining the Multiple Extended-target Lost-Track ratio (MELT$_{\tau_{ltr}}$):

$$\text{MELT}_{\tau_{ltr}} = \frac{1}{{}^{g}U} \sum_{d=1}^{{}^{g}U} \lambda_d^{\tau_{ltr}}, \tag{4.8}$$

which provides tracking performance at τ_{ltr} such that MELT$_{\tau_{ltr}} \in [0,1]$. The lower MELT$_{\tau_{ltr}}$, the better the performance. In order to enable the analysis of tracking performance at different accuracy levels, we compute MELT$_{\tau_{ltr}}$ for different τ_{ltr} values (Fig. 4.3). While the computation of MELT$_{\tau_{ltr}}$ may be useful from an application viewpoint, the performance comparison among trackers can be facilitated by providing the single-score average tracking performance which is generated as

$$\text{MELT} = \frac{1}{\Upsilon_{\tau_{ltr}}} \sum_{\tau_{ltr} \in \mathbb{R}_{(0,1]}} \text{MELT}_{\tau_{ltr}}. \tag{4.9}$$

The performance of a tracker at a particular accuracy level, τ_{ltr}, can be analysed by graphically showing the probability density function, $\mathcal{H}_{\tau_{ltr}}$, of the lost-track-ratio values (i.e. the values in the column τ_{ltr} of the Λ-matrix (Eq. 4.7)). Each sample of $\mathcal{H}_{\tau_{ltr}}$ represents the percentage of tracks with a particular lost-track-ratio (bin) at a specific value of τ_{ltr}. Bins are the equal-width intervals created by dividing the range of $\lambda_d^{\tau_{ltr}}$, where $\lambda_d^{\tau_{ltr}} \in [0,1]$. Fig. 4.3 shows examples of $\mathcal{H}_{\tau_{ltr}}$ plotted while varying the τ_{ltr} values. The higher the concentration of $\lambda_d^{\tau_{ltr}}$ values towards bin zero, the better the corresponding tracking performance at τ_{ltr}.

Fig. 4.3(a) shows an ideal tracking result with zero lost-track ratio value for all ${}^{g}\mathsf{T}_d$ at all τ_{ltr}. Similarly, Fig. 4.3(b) is the worst tracking result: the lost-track ratio is 1 for all ${}^{g}\mathsf{T}_d$ at all τ_{ltr}. Figures 4.3(c), 4.3(d) show the results of the Conditional Random Field based tracker (CRFBT) [181] and the Dynamic Programming-Non-Maxima Suppression based tracker (DP-NMS) [132] on ETH Sunnyday [7] using MELT and MOTP. MELT considers CRFBT to be better than DP-NMS and this can be seen from the highest concentration of values of CRFBT in the bins towards zero in Fig. 4.3(c). Consequently, MELT$_{\tau_{ltr}}$ values of CRFBT and DP-NMS computed for the variation of τ_{ltr} (Fig. 4.4(c)) show that CRFBT outperforms DP-NMS. The

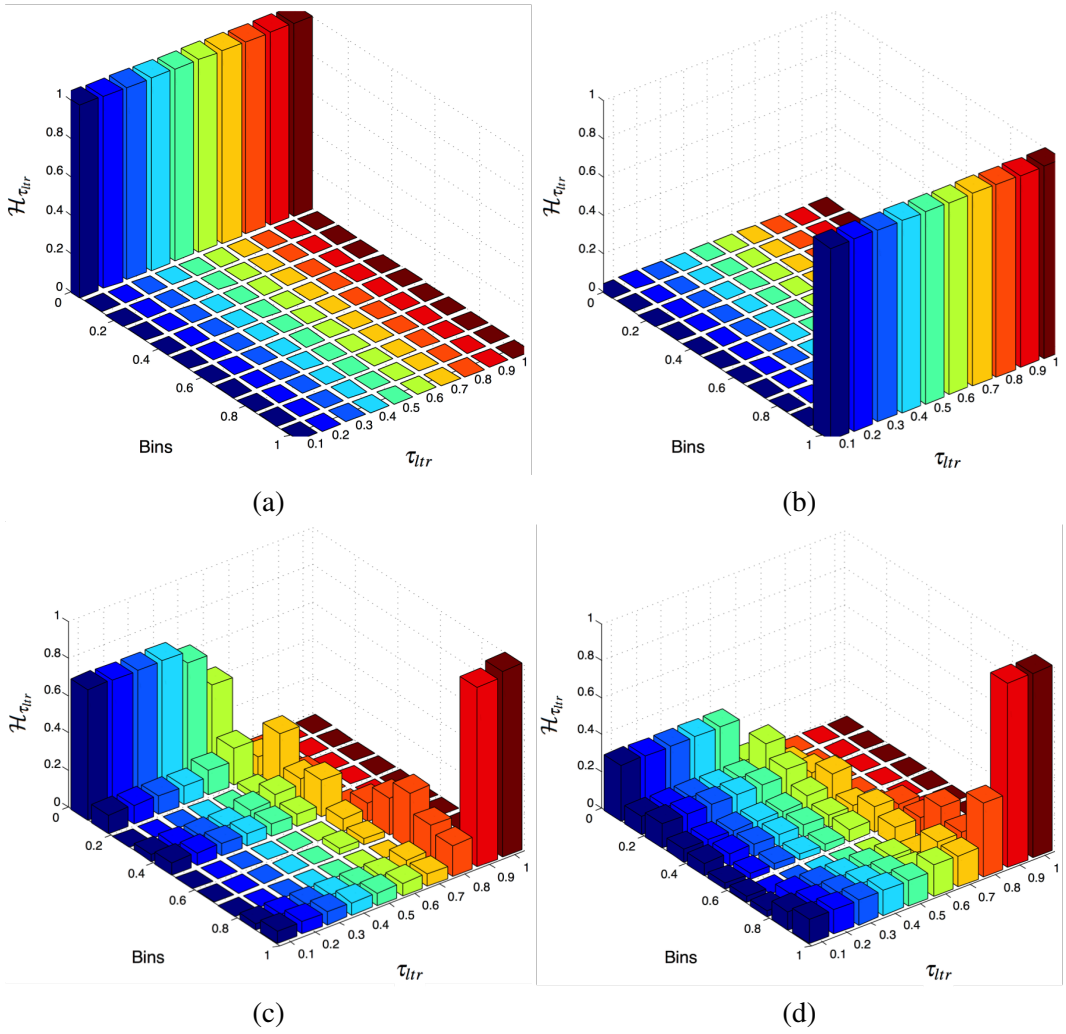


Figure 4.3: The probability density function $\mathcal{H}_{\tau_{ltr}}$ for a variation of τ_{ltr} values. (a) Ideal tracking result: the lost-track ratio is zero for all tracks at all the values of τ_{ltr}; hence, MELT $= 0$. (b) Worst tracking result: the lost-track ratio is 1 for all tracks at all values of τ_{ltr}; hence, MELT $= 1$. (c-d) MELT and MOTP of the Conditional Random Field based tracker (CRFBT) [181] and the Dynamic Programming-Non-Maxima Suppression based tracker (DP-NMS) [132] on ETH Sunnyday [7]; (c) CRFBT: MELT=0.39, MOTP=0.75; (d) DP-NMS: MELT=0.56, MOTP=0.77.

values of MELT$_{\tau_{ltr}}$ of CRFBT are lower for all τ_{ltr} than those of DP-NMS, meaning lower lost-track-ratio values and better tracking accuracy. On the other hand, MOTP ranks the performance of two trackers differently (i.e. opposite) because it does not take into account the overlap values of the estimated and ground-truth track pairs that are smaller than τ_{TP} (see Sec. 2.5.4), thereby not including the complete tracking accuracy in the assessment. MELT provides a holistic performance assessment taking into account all of the tracking information.

MELT also summarises tracking performance at different accuracy levels and provides an insight for analysis. For example, consider the MELT$_{\tau_{ltr}}$ plots of DP-NMS and the multi-target track-before-detect (MT-TBD) tracker [J2] shown in Fig. 4.4(c). MELT$_{\tau_{ltr}}$ shows that for $\tau_{ltr} <$

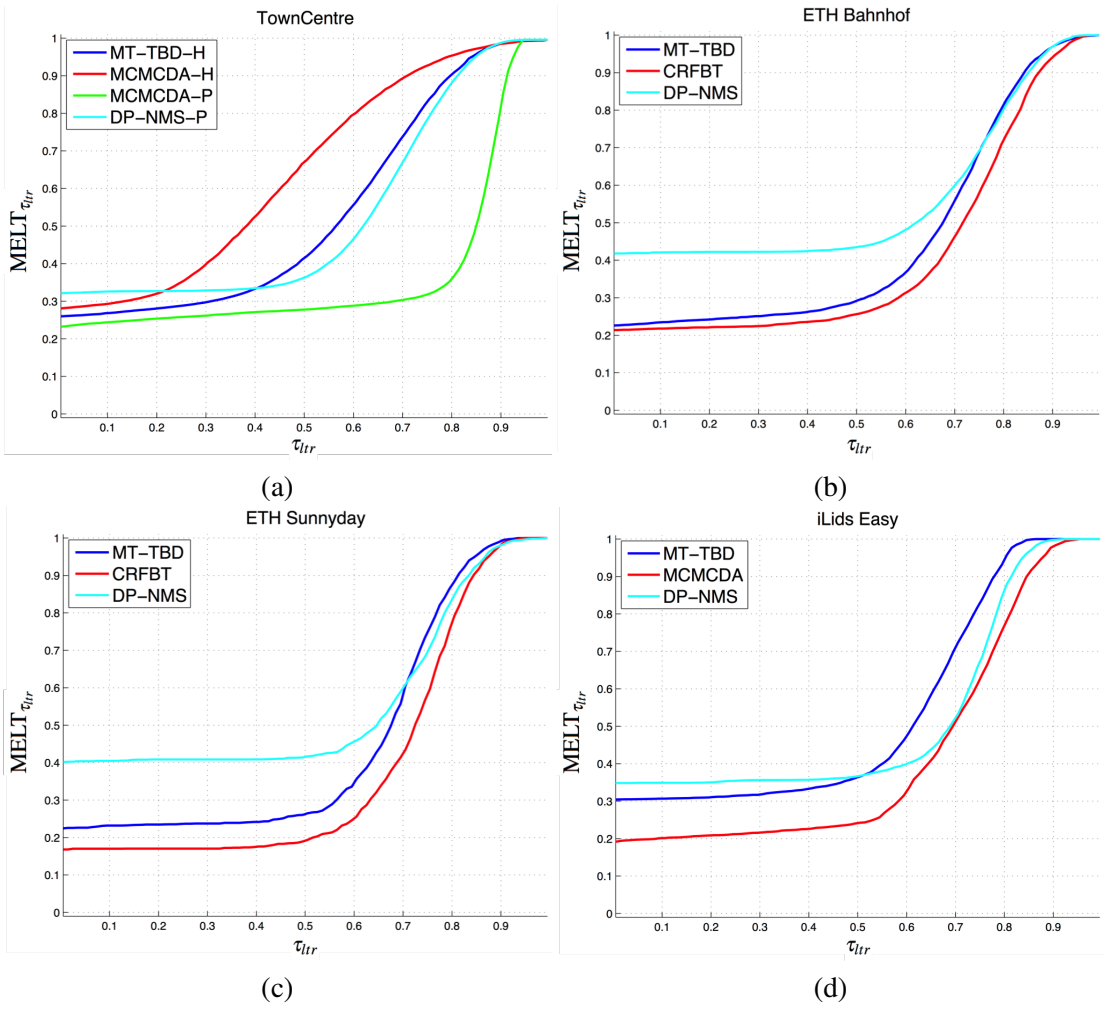


Figure 4.4: Evaluation of trackers' results at varying levels of accuracy (τ_{ltr}) using MELT$_{\tau_{ltr}}$ on all sequences. (a) MELT$_{\tau_{ltr}}$ of trackers on TownCentre sequence. 'H' and 'P' in the legend indicate the use of a tracker for head or person tracking, respectively; (b) MELT$_{\tau_{ltr}}$ of trackers on ETH Bahnhof sequence; (c) MELT$_{\tau_{ltr}}$ of trackers on ETH Sunnyday; and (d) MELT$_{\tau_{ltr}}$ of trackers on iLids Easy sequence.

0.72 (approx.), MT-TBD outperforms DP-NMS, after which DP-NMS outperforms MT-TBD. This analysis can be useful in selecting between these two trackers for an application that requires tracking with average overlap (accuracy) of e.g. 80%: DP-NMS would be a more suitable choice than MT-TBD.

4.2.3 Normalised ID changes

The Normalised ID Changes (NIDC) measure evaluates the ID changes taking into account the track duration in which they occur. In the case of a comparison of trackers producing tracks of different lengths, the normalisation of ID changes is preferable to simply counting the ID changes. Such quantification emphasises the long-term tracking ability with unique IDs of track-

ers. Moreover, since the score is normalised it can be more useful than the number of ID changes comparing trackers across different datasets. Unlike IDC [187] and MOTA [81], NIDC is parameter independent since its assignment solution is calculated as in Eq. 4.1.

Let ${}^{g}U_{IDC}$ be the number of ground-truth tracks with at least one ID change and

$$\text{NIDC}_d = \frac{|IDC_d|}{\text{IDC}_d^{max}} \tag{4.10}$$

be the $NIDC_d$ value for ground-truth track d and IDC_d^{max} the maximum number of ID changes that can occur for ground-truth track d (i.e. the length of track d). NIDC_d includes a contribution of ID changes for track d that is scaled by IDC_d^{max}, which is proportional to the duration of track d. This penalises the ID changes by the length of the track in the estimation of NIDC, instead of simply relying on counting ID changes [24, 81, 187]. NIDC quantifies the number of ID changes corresponding to all ground-truth tracks of the sequence:

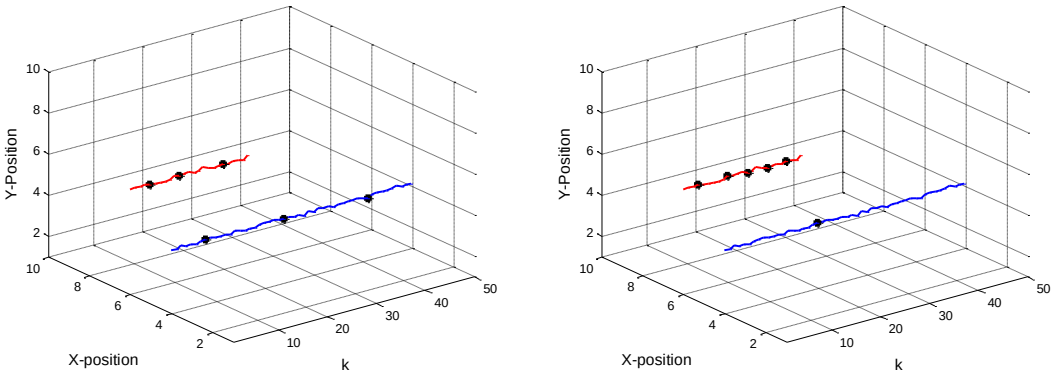
$$\text{NIDC} = \frac{1}{{}^{g}U_{IDC}} \sum_{d=1}^{{}^{g}U} \text{NIDC}_d, \tag{4.11}$$

where $\text{NIDC} \in [0,1]$. The lower NIDC, the better the performance in terms of ID maintenance.

Figure 4.5 shows two examples that compare NIDC with TF [24] and IDC [187]. The red ground-truth track (ID=1) and the blue ground-truth track (ID=2) shown in Fig. 4.5(a) have different lengths ($\text{IDC}_1^{max} = 25$ and $\text{IDC}_2^{max} = 50$) but have the same number of ID changes ($|IDC_1| = |IDC_2| = 3$). $\text{NIDC}_1 = 0.12$ is larger than $\text{NIDC}_2 = 0.06$ since the measure penalises the red track (shorter length) for the occurrence of the same number of ID changes. Unlike NIDC, TF does not distinguish these two cases as $\text{TF}_1 = 3$ and $\text{TF}_2 = 3$, as it does not consider track length. Both NIDC and TF are able to distinguish ID changes of two tracks in Fig. 4.5(b) as is shown in their listed values. Moreover, the ID changes of two different trackers are shown for the same sequence in Fig. 4.5(a) and Fig. 4.5(b), respectively. While IDC does not distinguish the results of the two trackers as $\text{IDC} = 6$ for both of them, NIDC differentiates between them ($\text{NIDC} = 0.09$ for (a) and $\text{NIDC} = 0.11$ for (b)).

4.3 Results and analysis

In this section, we validate the effectiveness of the proposed measures by comparing them with state-of-the-art measures. We then compare the performance of MT-TBD (Chapter 3) with state-



$\text{NIDC}_1 = 0.12$, $\text{TF}_1 = 3$
$\text{NIDC}_2 = 0.06$, $\text{TF}_2 = 3$
$\text{NIDC} = 0.09$, $\text{IDC} = 6$
(a)

$\text{NIDC}_1 = 0.20$, $\text{TF}_1 = 5$
$\text{NIDC}_2 = 0.02$, $\text{TF}_2 = 1$
$\text{NIDC} = 0.11$, $\text{IDC} = 6$
(b)

Figure 4.5: Comparison of the proposed Normalised ID Changes (NIDC) measure with Track Fragmentation (TF) [24] and ID Changes (IDC) [187]. (a) and (b) present results of two different trackers in terms of ID changes on the same sequence. Each example shows two ground-truth tracks; ID=1: red ground-truth track; ID=2: blue ground-truth track. ID changes are shown as black dots. (a) The length of the red track ($\text{IDC}_1^{max} = 25$) is shorter than that of the blue track ($\text{IDC}_2^{max} = 50$) and $|IDC_1| = |IDC_2| = 3$. Thus, $\text{NIDC}_1 = 0.12$ and $\text{NIDC}_2 = 0.06$ penalise the red track (shorter) more for the occurrence of the same number of ID changes as the blue track. However, $\text{TF}_1 = \text{TF}_2 = 3$ considers both the cases to be the same. (b) NIDC and TF can distinguish the different ID changes of the two tracks. The IDC measure considers (a) and (b) as the same cases since $\text{IDC} = 6$ for both, whereas NIDC can distinguish (a) and (b).

of-the-art trackers on real-world publicly-available datasets.

4.3.1 Experimental setup

We use four real-world datasets, namely TownCenter [20], ETH Bahnhof [7], ETH Sunnyday [7] and iLids Easy [5]. Details about TownCentre, and iLidsEasy are provided in Sec. 3.5.1. ETH Bahnhof and Sunnyday, recorded from a human-height moving camera, are composed of 999 and 354 frames, respectively, with a frame size of 640×480 recorded at 14 fps. The ground truth of Bahnhof has 95 person-tracks with an average of eight people per frame, while that of Sunnyday has 30 person-tracks with an average of five people per frame.

The trackers used in the experimental comparison include a combination of the Kanade-Lucas-Tomasi tracker [168] with Markov-Chain Monte-Carlo Data Association (MCMCDA) algorithm [20], a data association algorithm with the online learned Conditional Random Field Based Tracker (CRFBT) [181] and the Dynamic Programming Non-Maxima Suppression based tracker (DP-NMS) [132]. Tracking includes head and person (full-body) tracks from both static

and moving cameras. DP-NMS is tested on TownCentre, ETH Bahnhof and Sunnyday, and iLids Easy sequences for person tracking. MT-TBD is used for head tracking on the TownCentre sequence and for person tracking on the ETH Bahnhof and Sunnyday, and iLids Easy sequences. MCMCDA is used for head tracking on TownCentre and for person tracking on TownCentre and iLids Easy sequences. CRFBT is tested on ETH Bahnhof and Sunnyday sequences for person tracking. For the computation of N-MODA (Sec. 2.5.4), we use $\tau_{FP} = 0.50$ in the case of person tracking and $\tau_{FP} = 0.25$ in the case of head tracking, as done in [20].

The parameter values of the state-of-the-art trackers are those used in the original papers. The parameters for MT-TBD, in TownCentre and iLids Easy, are listed in Sec. 3.5.2. In the case of ETH datasets, we set $q_1 = 2$ and $\upsilon_{max} = 14$ since the datasets are recorded with a moving camera and at low frame rate, the displacement for walking people is larger than TRECVID and iLids Easy. We set $I_{min} = I_{max} = 2$ with noise $q_2 = 10^{-5}$, $\Sigma = 4$, $\sigma_1 = 0.80$, $\sigma_2 = 0.22$ and $\varsigma = 0.05$, as in TownCentre due to the provided thresholded confidence maps. $\alpha_1 = 0.2$ and $\alpha_2 = 0.02$ as for all the datasets.

4.3.2 Comparison of measures

We compare the proposed METE, MELT and NIDC measures with N-MODA, MOTP and IDC, respectively. Table 4.1 shows the scores of all measures obtained for all trackers.

The evaluation results using METE and N-MODA on TownCentre with head tracking (TownCentre-H) and with person tracking (TownCentre-P), and on Sunnyday show an agreement between both measures in terms of the relative ranking of trackers. However, there are disagreements on Bahnhof and iLids Easy. On Bahnhof, N-MODA of DP-NMS and MT-TBD are the same. This is because the normalisation in N-MODA formulation (Eq. 2.19) is with respect to the number of false positives and false negatives of tracking only and it does not consider the number of true positives. Since the total number of false positives and false negatives for DP-NMS (3525) and MT-TBD (3514) is comparable, their N-MODA is comparable. Although, the number of true positives for DP-NMS and MT-TBD are 5030 and 6222, respectively, METE ranks MT-TBD higher than DP-NMS since it implicitly takes into account true positives, false positives and false negatives. On iLids Easy, N-MODA ranks MT-TBD as the best tracker, which is not consistent with METE, which ranks MCMCDA as the best. N-MODA shows the best performance for MT-TBD because the total number of its false positives and false negatives (3639) is smaller than that of DP-NMS (3843) and MCMCDA (3698). METE, as discussed above, ranks their perfor-

Table 4.1: Overall comparison of trackers on different datasets with different evaluation measures. The coloured cells indicate the tracker's performance: the darker the colour, the better the performance. Key: TownCentre-H: Head tracking performed on TownCentre sequence; TownCentre-P: Person tracking performed on TownCentre sequence; METE: Multiple Extended-target Tracking Error; MELT: Multiple Extended-target Lost Track ratio; NIDC: Normalised ID Changes; AER: Accuracy Error Rate; CER: Cardinality Error Rate; MLT: Mean Length of ground-truth Tracks having id change(s); N-MODA: Normalised Multiple Object Detection Accuracy; MOTP: Multiple Object Tracking Precision; IDC: ID Changes; μ: mean value over the sequence; σ: standard deviation of values over the sequence in the case of METE, and standard deviation of values of accuracy error ($\mathcal{A}$) and cardinality error ($\mathcal{C}$) over the sequence for AER and CER, respectively.

Tracker	Dataset	METE $\mu(\sigma)$	MELT	NIDC	AER (σ)	CER (σ)	N-MODA	MOTP	IDC	MLT
MT-TBD [J2]	TownCentre-H	0.53 (0.08)	0.54	0.031	6.82 (2.54)	2.14 (1.92)	0.55	0.64	1798	320.00
MCMCDA [20]		0.62 (0.07)	0.65	0.038	8.48 (2.74)	1.82 (1.62)	0.46	0.51	1913	330.12
DP-NMS [132]	TownCentre-P	0.48 (0.08)	0.53	0.043	5.06 (1.52)	2.67 (2.02)	0.58	0.71	2637	321.61
MCMCDA [20]		0.33 (0.09)	0.37	0.030	3.64 (1.54)	1.81 (1.62)	0.62	0.80	1519	336.44
DP-NMS [132]	ETH Bahnhof	0.53 (0.13)	0.57	0.039	1.45 (0.69)	3.07 (1.85)	0.58	0.75	229	109.92
MT-TBD [J2]		0.44 (0.12)	0.46	0.050	2.42 (1.19)	1.56 (1.34)	0.58	0.75	307	103.51
CRFBT [181]		0.39 (0.12)	0.42	0.035	1.99 (0.86)	1.49 (1.26)	0.68	0.77	158	124.91
DP-NMS [132]	ETH Sunnyday	0.44 (0.11)	0.56	0.042	1.16 (0.55)	1.34 (0.93)	0.66	0.77	43	68.68
MT-TBD [J2]		0.47 (0.11)	0.46	0.041	1.60 (0.57)	1.09 (0.84)	0.61	0.73	56	91.50
CRFBT [181]		0.46 (0.12)	0.39	0.028	1.46 (0.52)	1.06 (0.78)	0.63	0.75	31	82.20
DP-NMS [132]	iLids Easy	0.40 (0.26)	0.52	0.011	0.40 (0.36)	0.65 (0.86)	0.60	0.74	104	632.87
MT-TBD [J2]		0.53 (0.22)	0.54	0.007	0.50 (0.36)	0.96 (1.10)	0.63	0.70	54	632.87
MCMCDA [20]		0.36 (0.26)	0.43	0.029	0.51 (0.45)	0.51 (0.76)	0.62	0.75	227	605.06

mance effectively by considering also the true positives (in addition to false positives and false negatives) of 6632, 6705 and 7969 for DP-NMS, MT-TBD and MCMCDA, respectively.

While MELT and MOTP agree on their relative ranking of trackers on TownCentre-H and TownCentre-P, they disagree on the remaining sequences (Tab. 4.1). In the case of Bahnhof, MOTP of MT-TBD and DP-NMS are the same, whereas MELT ranks MT-TBD higher than DP-NMS. The MELT$_{\tau_{ltr}}$ plots also show a better performance of MT-TBD for most of the variations of τ_{ltr} than DP-NMS (Fig. 4.4(b)). The disagreement of MOTP is due to its dependence on the threshold value τ_{TP}. MOTP considers only the overlap values of pairs greater than τ_{TP}, which may lead to the exclusion of some tracking information in the performance assessment. On the other hand, MELT uses all of the tracking information in the performance assessment to present a comprehensive performance evaluation that can more effectively reflect the trackers' comparison. In the case of Sunnyday, there is a disagreement between MELT and MOTP in selecting the best tracker, as already discussed in Sec. 4.2.2. In the case of iLids Easy, MOTP of DP-NMS and MCMCDA are comparable; however, based on their MELT scores and MELT$_{\tau_{ltr}}$ plots (Fig. 4.4(d)), the difference in their performance is clear. The inconsistencies of MOTP in Sunnyday and iLids Easy are due to its parameter dependency.

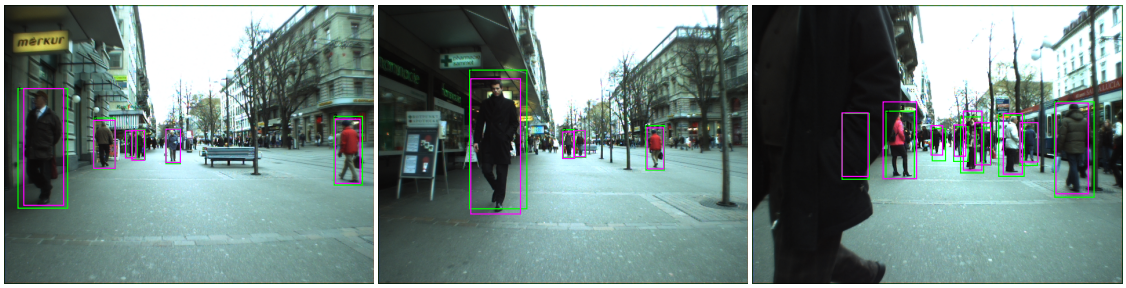
NIDC and IDC agree in their relative evaluation of trackers on TownCentre, Bahnhof and

iLids Easy (Tab. 4.1). The effectiveness of NIDC can be noticed in the case of Sunnyday. IDC considers the performance of DP-NMS to be better than MT-TBD. NIDC shows a slightly better performance for MT-TBD than DP-NMS despite the fact that the former has produced more ID changes than the latter. This is because NIDC provides ID evaluation while considering also the track length. Since MLT (mean length of ground-truth tracks having ID change(s)) of the MT-TBD is much higher than DP-NMS, NIDC penalises the ID changes of the former less.

Figure 4.6 shows the evaluation results of CRFBT on key frames of Bahnhof using METE and MODA. All the targets are tracked in the results shown in Fig. 4.6(a), (b) and (c). The value of METE increases from (a) to (c) because of the decrease in the amount of overlap (lower accuracy) among the associated pairs of estimated and ground-truth bounding boxes. The cases shown in Fig. 4.6(d) and (e) have $\mathcal{C} = 1$; however, METE in (e) is higher than that in (d). In Fig. 4.6(f), 79% of targets are correctly tracked (11 out of 14), hence its METE value (0.400) is higher than that in (e) where 90% of targets are correctly tracked with a METE value of 0.373. In Fig. 4.6(g), the percentage of tracked targets reduces further to 73%, hence its METE value is higher than (f). Although the percentage of tracked targets in the case of Fig. 4.6(h) (75%) is higher than (g), METE is slightly higher in the case of the former because of a more inaccurate overlap in the case of (h). Likewise, METE for the cases shown in Fig. 4.6(i-l) is influenced by the corresponding accuracy and cardinality errors. MODA does not distinguish among the cases in Fig. 4.6(a-c) (MODA=1) despite the difference in their respective overlaps. This insensitivity of MODA is due to the threshold (τ_{TP}) used to determine false negatives and false positives (Eq. 2.20). Another point to highlight is the disagreement between METE and MODA in the cases shown in Fig. 4.6(h) and (i). Unlike METE, MODA considers the case in (i) to be better than (h). This is because in the case of (h), MODA considers 58% (7 out of 12) of estimated bounding boxes to be correctly associated to those of the ground truth, excluding the third and the sixth pairs (starting from the right) that are not considered to be valid associations since their overlap is below τ_{TP}. Differently, METE, being independent of thresholds, considers these two pairs in the evaluation of the score and penalises them appropriately. In the case of Fig. 4.6(i), 66% of the ground-truth targets are correctly associated and there is one false positive, hence the MODA value is higher for this case.

4.3.3 Comparison of trackers

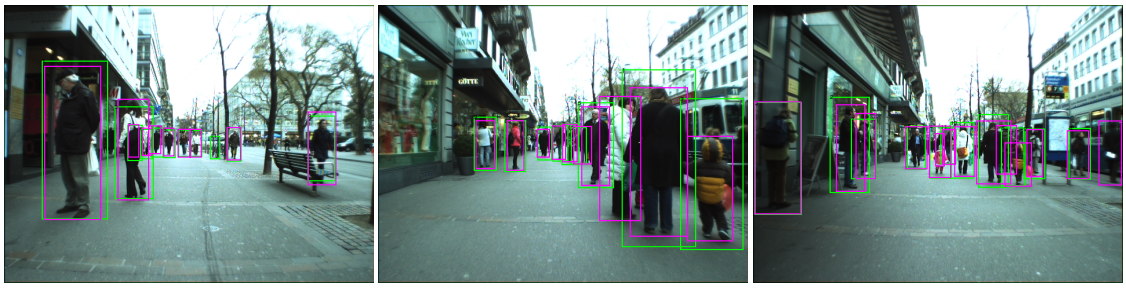
We evaluate MT-TBD while analysing the effectiveness of the proposed measures.



(a) k=194, ${}^{g}u$=6
$\mathcal{A}$=0.79, $\mathcal{C}$=0
METE=0.13, MODA=1

(b) k=29, ${}^{g}u$=4
$\mathcal{A}$=0.69, $\mathcal{C}$=0
METE=0.17, MODA=1

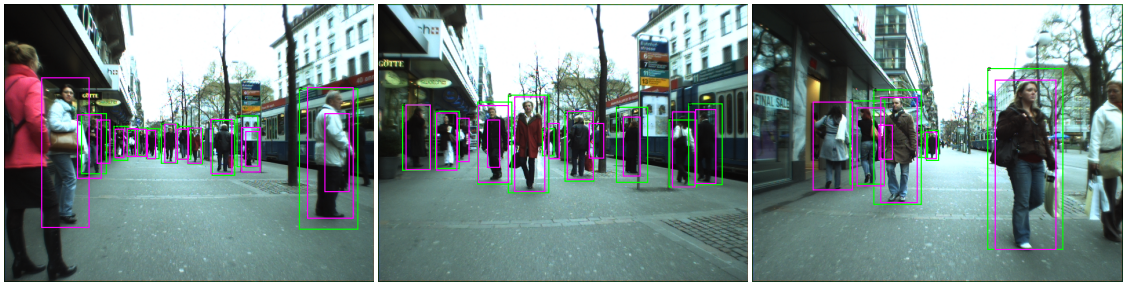
(c) k=616, ${}^{g}u$=9
$\mathcal{A}$=2.27, $\mathcal{C}$=0
METE=0.25, MODA=1



(d) k=860, ${}^{g}u$=10
$\mathcal{A}$=2.25, $\mathcal{C}$=1
METE=0.30, MODA=0.90

(e) k=587, ${}^{g}u$=10
$\mathcal{A}$=2.73, $\mathcal{C}$=1
METE=0.37, MODA=0.90

(f) k=513, ${}^{g}u$=14
$\mathcal{A}$=2.60, $\mathcal{C}$=3
METE=0.40, MODA= 0.79



(g) k=665, ${}^{g}u$=15
$\mathcal{A}$=3.35, $\mathcal{C}$=4
METE=0.49, MODA=0.73

(h) k=704, ${}^{g}u$=12
$\mathcal{A}$=2.95, $\mathcal{C}$=3
METE=0.50, MODA= 0.42

(i) k=952, ${}^{g}u$=6
$\mathcal{A}$=2.17, $\mathcal{C}$=1
METE=0.53, MODA=0.50

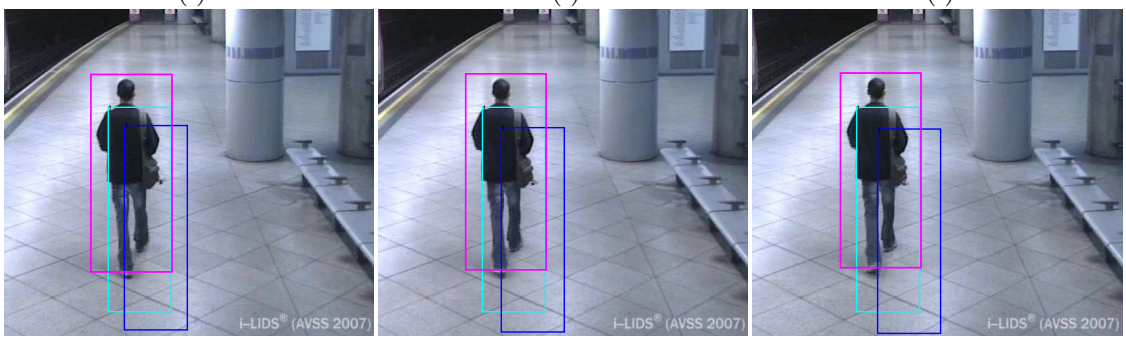
(j) k=724, ${}^{g}u$=13
$\mathcal{A}$=3.62, $\mathcal{C}$=4
METE=0.59, MODA=0.39

(k) k=596, ${}^{g}u$=11
$\mathcal{A}$=1.35, $\mathcal{C}$=6
METE=0.67, MODA=0.46

(l) k=764, ${}^{g}u$=8
$\mathcal{A}$=1.93, $\mathcal{C}$=4
METE=0.74, MODA= 0.25

Figure 4.6: Evaluation of the results of CRFBT on Bahnhof sequence using METE and MODA. Subscript k is removed from the variables for simplicity in the notation. Ground truth and tracker's estimates are shown as magenta and green bounding boxes, respectively. Results are ordered in terms of ascending METE values.

(a) Cyan: $O(\cdot) = 0.57$ Blue: $O(\cdot) = 0.56$ (b) Cyan: $O(\cdot) = 0.57$ Blue: $O(\cdot) = 0.41$ (c) Cyan: $O(\cdot) = 0.56$ Blue: $O(\cdot) = 0.36$



(d) Cyan: $O(\cdot) = 0.56$ Blue: $O(\cdot) = 0.32$ (e) Cyan: $O(\cdot) = 0.55$ Blue: $O(\cdot) = 0.29$ (f) Cyan: $O(\cdot) = 0.55$ Blue: $O(\cdot) = 0.27$

Figure 4.7: Example showing the limitation of Multiple Object Tracking Precision (MOTP) [81]. Cyan tracker: MOTP=0.56, MELT=0.45; Blue tracker: MOTP=0.56, MELT=0.64. Unlike the proposed measure MELT, MOTP does not distinguish two tracking results due to its parameter dependence. Magenta: ground truth.

On TownCentre-H, MT-TBD outperforms MCMCDA using mean METE, MELT, NIDC and AER (see r.[2] 1, 2 in Tab. 4.1), which is also confirmed in the MELT$_{\tau_{ltr}}$ plots (Fig. 4.4(a)). MT-TBD has a better NIDC than MCMCDA because of its better ID management mechanism, which involves minimising the mixing of target particles in the Bayesian state estimation [J2]. Interestingly, CER differs from the remaining measures and shows better performance for MCMCDA compared to MT-TBD. The higher CER of MT-TBD is due to a greater number of tracking failures or missed targets. Since AER is lower for MT-TBD, this points to fewer occurrences of tracking failures than missed targets.

On TownCentre-P, MCMCDA outperforms DP-NMS based on mean METE, MELT, NIDC, AER and CER (see r. 3, 4 in Tab. 4.1). It is also interesting to highlight the clear improvement in the evaluation results of MCMCDA using the proposed measures on TownCentre-P compared to TownCentre-H, which is inline with the results of the original paper [20].

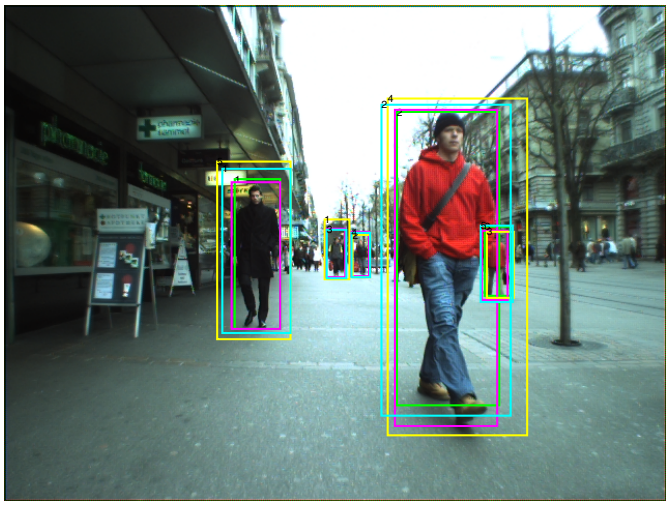
[2]'r' refers to the row number in Tab. 4.1 not considering the row with titles.

On Bahnhof, mean METE, MELT, NIDC and CER rank CRFBT as the best tracker compared to DP-NMS and MT-TBD (see r. 5, 6, 7 in Tab. 4.1). This is also visible in the $\text{MELT}_{\tau_{ltr}}$ plots (Fig. 4.4(b)). The reason for the best NIDC of CRFBT is its capability to address ID changes using motion and appearance 'affinities' [181], enabling it to distinguish and separate nearby targets. There is an inconsistency in the case of AER that ranks CRFBT as second-best tracker after DP-NMS. Furthermore, the CER of DP-NMS is almost twice that of MT-TBD and CRFBT. This is due to the limited capability of DP-NMS, unlike MT-TBD and CRFBT, to link fragmented tracks that increases the cardinality error. The fragmentations in the case of DP-NMS are caused by worse handling of long-term occlusions compared to MT-TBD and CRFBT (Fig. 4.8).

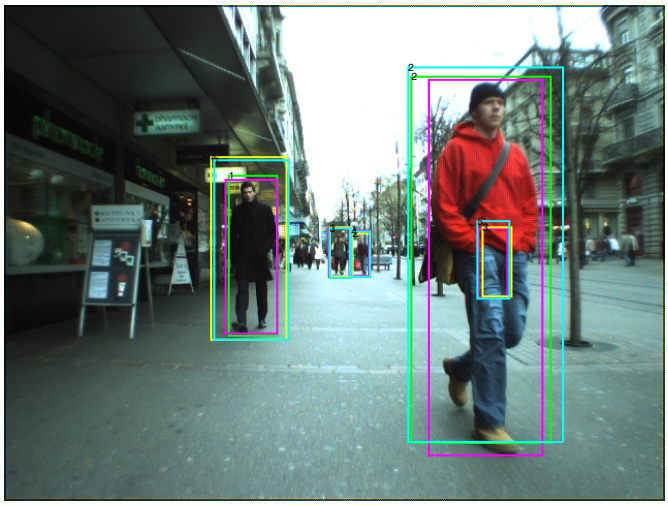
On Sunnyday, we compare MT-TBD to DP-NMS and CRFBT as on Bahnhof. Some inconsistencies can be noticed in the evaluation results on Sunnyday compared to those on Bahnhof. Firstly, unlike on Bahnhof, the evaluation based on mean METE on Sunnyday shows a better performance of DP-NMS compared to MT-TBD and CRFBT (see r. 8, 9, 10 in Tab. 4.1). This is probably because the person detector [51] used with DP-NMS can better deal with the higher scene brightness in Sunnyday than the detector [179] used with MT-TBD and CRFBT, which results in the improved tracking performance of DP-NMS. Secondly, unlike on Bahnhof, NIDC of MT-TBD is better than DP-NMS on Sunnyday despite the fact that IDC of the former is higher than the latter in both sequences.

On iLids Easy, the evaluation of trackers using mean METE and MELT shows the superior performance of MCMCDA compared to DP-NMS and MT-TBD (see r. 11, 12, 13 in Tab. 4.1). The superior mean METE and MELT of MCMCDA over DP-NMS is consistent with their mean METE and MELT on TownCentre-P. Moreover, the analysis of $\text{MELT}_{\tau_{ltr}}$ plots (Fig. 4.4(d)) provides an interesting insight about the performance of MT-TBD and DP-NMS, revealing that $\text{MELT}_{\tau_{ltr}}$ of MT-TBD is better than DP-NMS for $\tau_{ltr} < 0.5$ and the reverse is true thereafter. This suggests that DP-NMS is a more suitable choice for tracking with higher accuracy and MT-TBD should be preferred with lower accuracy since its lost-track-ratio values are smaller at lower τ_{ltr}. Additionally, while CER of DP-NMS is the highest on the rest of the sequences, MT-TBD has the highest CER on iLids Easy. Furthermore, the best NIDC of MT-TBD on iLids Easy is due to its better ID management ability as discussed earlier. Interestingly, although MLT of MT-TBD and DP-NMS is the same[3] (see r. 11, 12 in Tab. 4.1), the higher IDC of the latter leads to its

[3]The same MLT is because ID change(s) for both trackers have occurred in the same tracks.



(a) $k = 16$



(b) $k = 19$

(c) $k = 24$

Figure 4.8: Example of target occlusion in Bahnhof sequence. DP-NMS (green bounding box) loses the target due to occlusion in (b); however, the other trackers successfully handle it. Yellow: MT-TBD; cyan: CRFBT; magenta: ground truth.

inferior NIDC.

Table 4.1 also presents the variation in the performance of the trackers in terms of standard deviation (σ) values for different measures. In the case of METE, σ is comparable on all sequences. As for AER, while MCMCDA has the highest σ on TownCentre and iLids Easy (hence, the highest performance variation over time), MT-TBD has the highest performance variation over time on Bahnhof and Sunnyday. As for CER, the trend of the σ values of trackers on each dataset is the same as the trend of the corresponding CER values.

4.4 Summary

We proposed three measures (METE, MELT, NIDC) that quantify key factors in extended multi-target tracking: accuracy, cardinality and ID changes. These measures are parameter independent, numerically bounded and account for target-size changes. METE provides a holistic error assessment using an effective trade-off between accuracy and cardinality errors. MELT enables the analysis of tracking performance at varying accuracy levels that can facilitate the selection of trackers for specific applications. NIDC penalises ID changes as a function of the length of the track in which they occur. We presented an extensive experimental validation and comparison of these measures with the state-of-the-art measures on recent multi-target trackers using challenging real-world sequences. The proposed measures are suitable for targets that are modelled in terms of their position and 2D image-plane-occupied area, as commonly considered in the literature [81, 99, 124].

METE, MELT and NIDC mostly evaluated the performance of the trackers in the same way as MODA, MOTP and IDC, but with the advantage of being parameter independent and numerically bounded. The evaluation performed on head tracking and body tracking results did not require the setup of parameters and the ranking of the tracker's performance was still correct. We also showed that from METE and MELT it is possible to extract further details for a more in depth analysis of the performance. For example, by using the different levels of accuracy with $\text{MELT}_{\tau_{ltr}}$ we observed that in ETH Sunnyday CRFBT is the best overall, whereas MT-TBD is better than DP-NMS only for overlap values less than 0.7. Therefore, unlike METE, the dependence of MODA on the preset overlap threshold limits its ability to clearly distinguish different tracking results, and unlike MELT, the threshold dependency of MOTP may result in an inaccurate evaluation of tracking performance. Unlike IDS, NIDC evaluates the ID changes as a

function of the track length in order to better understand where the errors occur.

We then extensively compared the performance of MT-TBD with state-of-the-art multi-target trackers. CRFBT was the best tracker based on the evaluation of ID changes, followed by MT-TBD. CRFBT employes an optimisation algorithm for data association where colour features and the pairwise modelling of motion was key for the discrimination of targets during tracking. MT-TBD had better performance than MCMCDA and DP-NMS in terms of ID maintenance on targets. This suggests that MT-TBD produces less fragmented tracks and is more suitable for long-term tracking than MCMCDA and DP-NMS. The overall performance of MT-TBD in terms of METE and MELT was shown to be, on average, similar to that of the state-of-the-art trackers with which it was compared. However, the main differences between the four trackers is that CRFBT and DP-NMS are offline and they need the whole set of detections to extract trajectories, whereas MT-TBD and MCMCDA can run in a shifting temporal window and extract the trajectories on-the-fly. DP-NMS reported the lowest accuracy error and its cardinality error was generally the highest. DP-NMS was not able to handle occlusions.

In the next chapter, we present a multi-target tracking framework employed on scenes with a high density of targets. We evaluate the performance of different target tracking methods including MT-TBD (Chapter 3), a graph-based algorithm capable to handle point targets with challenging motion properties and state-of-the-art tracking methods.

Chapter 5

Tracking on low-SNR videos

5.1 Introduction

Target trajectories can be generated by temporally associating candidate locations with multi-target trackers [20, 73, 140, 159, 181] or directly from confidence maps [142, J2]. Common approaches address the problem of generating candidate target locations by applying thresholds, clustering and Non-Maxima Suppression (NMS) to the confidence values [44]. Target locations can also be directly computed by thresholding and clustering intensity values from images (target-intensity maps) [140]. Weakly-appearing targets (or barely detected parts) may lead to intensity values (or confidence values) with multiple peaks in the target area [51]. However, values with multiple peaks can be due to adjacent targets, and hence explicit models to separately detect targets need to be defined. In the case of high-density scenes with targets having the same or similar appearance, multi-target tracking is addressed with strong priors on target motion [90] and appearance [73].

In this chapter, we present a method for multi-target tracking on low-SNR videos that contain high density of compact targets (Fig. 5.1). We initially describe a novel method to extract target locations and a graph-based data association method for multi-target tracking. The detection method is applied on low-SNR videos and can deal with multi-peak intensities generated by adjacent targets. We use intensity gradient information and isocountours applied to target-intensity maps (see Sec. 1.1 for the definition of target-intensity maps), which enable us to extract candidate target locations without the need for trained target appearance models [183] or temporal

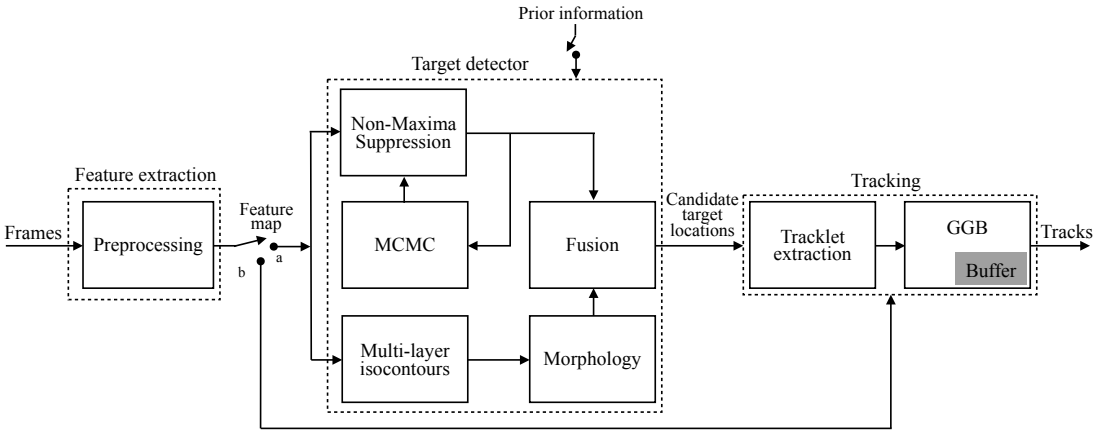


Figure 5.1: Overall block diagram for target detection and tracking. Tracking can be either performed using candidate target locations (switch is on "a") or on confidence maps (switch is on "b") (e.g. MT-TBD). Prior information (user intervention) can be used to improve the confidence detection. The pipeline of the proposed approach has the prior information input *off* and the switch on "a". Key. MCMC: Monte Carlo Markov Chain; GGB: Greedy-Graph Based.

dependencies that might lead to drifts in the case of overlapping targets [159]. We model compact targets with simple shape priors, such as size, and generate candidate target locations via local maxima searching with a probabilistic model to estimate the locations of targets in high-density scenes. We track the targets with a graph-based method that pair-wise matches short tracks [189] and performs the backward validation of them within the shifting temporal window [155]. The number of targets is implicitly inferred by the algorithm. Importantly, as for MT-TBD (Sec. 3), initialisation and termination of tracks are automatically performed and they can occur in any location of the scene, whereas with a network flow approach [189] start and end locations have to be defined a priori. MT-TBD is tested on this scenario and compared with the graph-based data association method, as well as alternative state-of-the-art methods. The proposed tracking algorithm outperforms alternative methods on challenging datasets.

The chapter is organised as follows. Sec. 5.2 describes the feature extraction method, which is devided into two parts, namely gradient-climbing based detector (Sec. 5.2.1) and hierarchical-isocontour based morphology (Sec. 5.2.2). Sec. 5.3 describes the graph-based data association method for multi-target tracking. The results and the comparisons with alternative methods are discussed in Sec. 5.4. Finally, in Sec. 5.5 we present a summary of the achievements.

5.2 Feature extraction and target detection

The detection of similar targets on high-density and low-SNR videos is performed by exploiting target-intensity maps using the intensity-gradient information and the intensity values at different

levels (isocontours) to generate $\mathcal{Z}_k$ (see Sec. 1.2), as discussed below.

Let C_k be a target-intensity map extracted from the frame v_k, having $C_{i,k}$ as elements with $i = 1, ..., I$ and I the total number of pixels in a frame. Each $C_{i,k} \in \mathbb{R}_{[0,1]}$ is the (feature) intensity value of the i^{th} pixel with generic coordinates (x_i, y_i). The larger $C_{i,k}$, the clearer the target representation. The b^{th} target detection $\mathbf{z}_{b,k}$ is represented as in Eq. 1.2:

$$\mathbf{z}_{b,k} = [x_{b,k} \;\; y_{b,k} \;\; S_{b,k} \;\; \iota_{b,k}]^T, \tag{5.1}$$

where $\iota_{b,k} = \sqrt{\sum_{i \in \mathcal{C}^{S_{b,k}}} C_{i,k}^2}$ is the energy calculated from the intensity map, with $\mathcal{C}^{S_{b,k}}$ being the set of intensity values within the region defined by $S_{b,k}$. T is the transpose operator of a matrix. Without loss of generality, we use an elliptical shape (Fig. 5.2(g,h)) [73, 181] and consider $S_{b,k} = (r_1, r_2, \theta_{b,k})$, where the scalar values r_1 and r_2 are the major and minor semi-axis, respectively, and $\theta_{b,k}$ is the orientation. Given $\mathcal{Z}_k$ for k=1, ..., K, tracking temporally associates detections to generate trajectories. $\mathcal{T} = \{\mathsf{T}_a\}_{a=1}^{A}$ is the set of temporally-ordered trajectories, where T_a is the a^{th} trajectory with an arbitrary duration and A is the total number of trajectories (Sec. 1.2). The smaller the index a, the earlier the starting frame of the corresponding trajectory.

5.2.1 Detector based on gradient climbing

Barely visible (low intensities) and close targets are often characterised by intensities with multiple peaks. We assume that each high intensity value of the target-intensity map is a candidate target detection and such a map usually contains broad peaks. The gradient of the (scalar) intensity values is used to localise and discriminate targets since it helps the enhancement of spatial gaps among nearby targets. In the case of partially-overlapping targets, it can also help the fitting of a prior shape model by exploiting visible parts of both the targets involved in the occlusion (the occluding and occluded target). The target localisation is then achieved by using an iterative algorithm that exploits such intensity values that, after low-pass filtering (e.g. Gaussian kernel), tend to increase towards the centre of the targets. We aim to find the best fit between the target shape and the target intensity, while disregarding distractors due to nearby targets and multi-peak intensities (e.g. region 1 in Fig. 5.2(g) and 5.2(m)).

The method begins by generating detections over the intensity map while discarding those detections located near local maxima. This step enables the reduction of the computational complexity of the subsequent steps. Since at this stage there is no knowledge about the orientation of

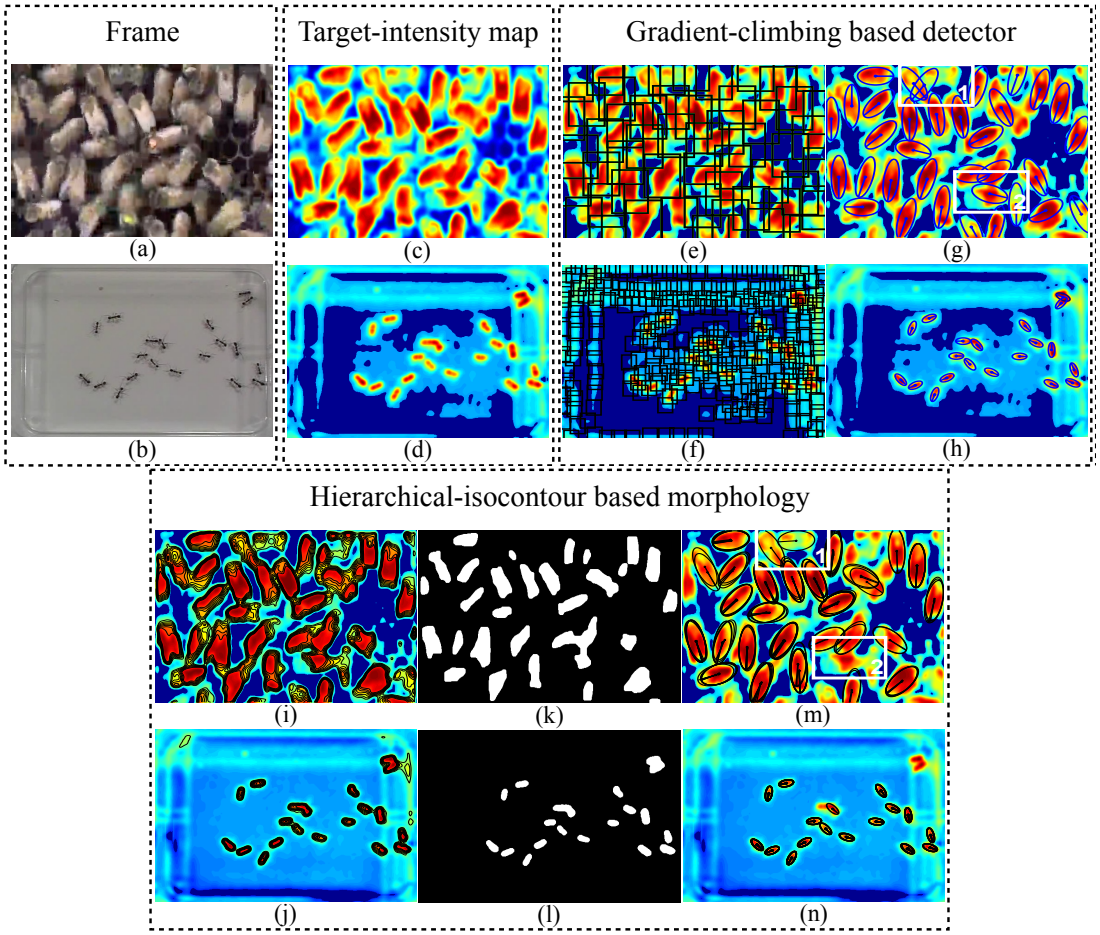


Figure 5.2: Detection process using the gradient-climbing based detector and hierarchical-isocontour based morphology: (a,b) input frames; (c,d) maps representing the enhanced target intensities (target-intensity map); (e,f) mid-level step where squared detections are initialised using the Non-Maxima Suppression algorithm on the target-intensity map; (g,h) resulting detections obtained with the proposed approach based on MCMC. (i,j) Multi-layer isocontours on the target-intensity maps; (k,l) hole filling and erosion followed by dilation applied on each layer; (m,n) detections obtained evaluating shape properties on each region. (g,m) Region 1 and region 2 highlight challenging cases that can be addressed by taking into account the advantages of the two detectors.

the targets and their location, detections are initialised with a square area for each pixel i (i.e. a single detection is initialised for each $C_{i,k}$ where the number of pixels I is large). The square area allows us to begin the detection process with a simple dummy shape, which is a computationally effective solution to get rid of a few candidate target locations with low intensity values. This process can be achieved by formulating dummy detections as $\mathbf{d}_{i,k} = [x_i\ y_i\ r_1\ C_{i,k}]^T$ at frame k, with $\mathcal{D}_k = \{\mathbf{d}_{i,k}\}_{i=1}^{I}$ and r_1 is the side of a squared area centred at (x_i, y_i). We use the Non-Maxima Suppression (NMS) algorithm [51, 132] to skim detections from the set $\mathcal{D}_k$ with an overlapping area greater than a value τ_{nms}. The set of *skimmed* detections is defined as $\tilde{\mathcal{Z}}_k^1 = \{\tilde{\mathbf{z}}_{j,k}^1\}_{j=1}^{J}$, with $J(\ll I)$ the number of detections surviving after NMS.

The skimmed detections $\tilde{\mathcal{Z}}_k^1$ are subsequently made to align on the actual target locations with MCMC [142], exploiting the prior shape information $S_{m,k}$. In the case of high densities of targets, MCMC can probabilistically reach equilibrium for a large-state spaces with an unknown distribution. Let $\mathcal{Z}_k^1 = \{\mathbf{z}_{m,k}^1\}_{m=1}^{M_k^1}$ be the subset of resulting detections $\mathbf{z}_{m,k}^1$ obtained after applying MCMC, where M_k^1 is the number of detections at k. $\mathbf{z}_{m,k}^1$ has the same elements as Eq. 5.1. Each $\mathbf{z}_{m,k}^1$ is associated to a distribution $p(\mathbf{z}_{m,k}^1)$, which is computed using the intensities of C_k with the intensities expected by the prior intensity distribution of a single target $\mathcal{P}(\mathbf{z}_{m,k}^1) = \mathcal{N}(\boldsymbol{\mu}_{\mathcal{P}}, \boldsymbol{\Sigma}_{\mathcal{P}})$, where $\boldsymbol{\mu}_{\mathcal{P}} = (x_{m,k}^1, y_{m,k}^1)$ is the mean location and the covariance $\boldsymbol{\Sigma}_{\mathcal{P}}$ is a function of the target shape $(r_1, r_2, \theta_{m,k}^1)$. Although in our case $\mathcal{P}(\cdot)$ follows a 2D Gaussian distribution, it can be substituted with another function in the case of different intensity distributions. Therefore, the detection alignment with MCMC is specifically performed iteratively by picking each $\tilde{\mathbf{z}}_{j,k}^1 \in \tilde{\mathcal{Z}}_k^1$, proposing a move and validating it using a likelihood function in order to create $\mathcal{Z}_k^1$. This can be achieved using Metropolis-Hastings (M-H) algorithm [142], which enables inference of the global distribution $p(\mathcal{Z}_k^1)$ by sampling from an unknown multi-dimensional distribution with states that have many dimensions. Let $\mathbf{z}_{m,k}^{1,h}$ define the h^{th} iteration (move) of a proposed detection and $\mathcal{H}$ the total number of iterations. The initialisation of M-H, i.e. h=0, is done for each m such that $\mathbf{z}_{m,k}^{1,0} = \tilde{\mathbf{z}}_{j,k}^1$. M-H moves the detection $\mathbf{z}_{m,k}^{1,h}$ to a new detection $\mathbf{z}_{m,k}^{1,h+1}$ using a proposal density $q(\mathbf{z}_{m,k}^{1,h+1} | \mathbf{z}_{m,k}^{1,h}, \mathrm{C}_k)$, only if $\gamma \leq \alpha$, where $\gamma \sim \mathcal{U}[0,1]$ and α is the acceptance probability

$$\alpha = \min\left(1, \frac{p(\mathbf{z}_{m,k}^{1,h+1}|\mathrm{C}_k)}{p(\mathbf{z}_{m,k}^{1,h}|\mathrm{C}_k)}\right), \tag{5.2}$$

with

$$p(\mathbf{z}_{m,k}^{1,h+1}|\mathrm{C}_k) = p(\mathrm{C}_k|\mathbf{z}_{m,k}^{1,h+1}) q(\mathbf{z}_{m,k}^{1,h+1}|\mathbf{z}_{m,k}^{1,h}, \mathrm{C}_k), \tag{5.3}$$

where $p(\mathrm{C}_k|\mathbf{z}_{m,k}^{1,h+1})$ is the likelihood function.

The proposal density $q(\cdot)$ defines the dynamic model

$$\mathbf{z}_{m,k}^{1,h+1} = F_{m,k}^h \mathbf{z}_{m,k}^{1,h} + w_{m,k}^h, \tag{5.4}$$

where $w_{m,k}^h \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_w)$ with $\boldsymbol{\Sigma}_w$ the covariance of the noise and $F_{m,k}^h$ is a linear transformation

dependent on iterations and time

$$F_{m,k}^{h}=\begin{bmatrix}\begin{bmatrix}1+\frac{\dot{x}_{m,k}^{1,h}}{x_{m,k}^{1,h}} & 0\\ 0 & 1+\frac{\dot{y}_{m,k}^{1,h}}{y_{m,k}^{1,h}}\end{bmatrix} & \mathbf{0}_{2\times 4}\\ \mathbf{0}_{4\times 2} & \mathbf{I}_{4\times 4}\end{bmatrix}, \tag{5.5}$$

where $(\dot{x}_{m,k}^{1,h},\dot{y}_{m,k}^{1,h})$ is a translation vector. This vector is defined as the spatial translation from $(x_{m,k}^{1,h},y_{m,k}^{1,h})$ towards the maximum value of the normalised cross-correlation [58] between $\mathcal{P}(\mathbf{z}_{m,k}^{1,h})$ and $C_{i,k}$ for i=1,...,I, within the squared domain $r_1\times r_1$ of $\mathbf{z}_{m,k}^{1,h+1}$. $\mathbf{0}_{\mathfrak{n}'\times\mathfrak{m}'}$ is a matrix of zeros and $\mathbf{I}_{\mathfrak{n}'\times\mathfrak{m}'}$ is the identity matrix with $\mathfrak{n}'$ rows and $\mathfrak{m}'$ columns.

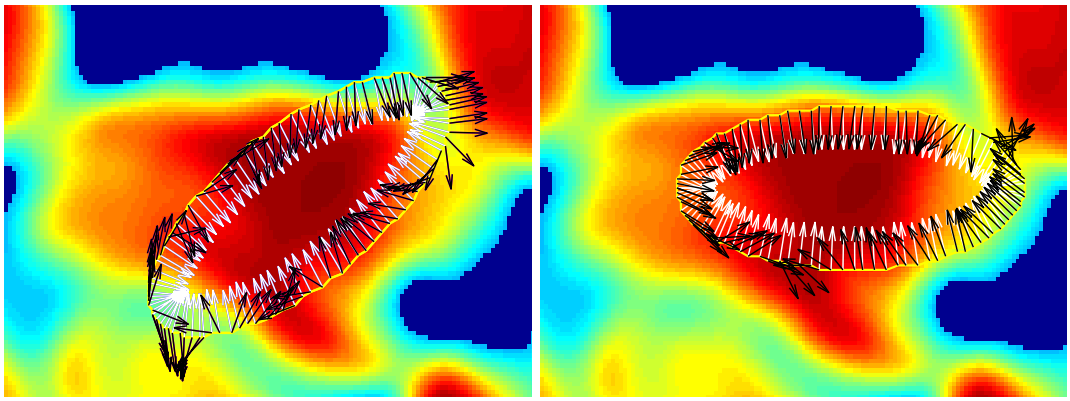
The likelihood function $p(\mathrm{C}_k|\mathbf{z}_{m,k}^{1,h+1})$ is calculated through Maximum A Posteriori (MAP) by varying the orientation $\theta_{m,k}^{1,h+1}$ within the interval Θ=$[0,\pi]$ of the translated detection $\mathbf{z}_{m,k}^{1,h+1}$. $p(\mathrm{C}_k|\mathbf{z}_{m,k}^{1,h+1})$ employs (i) the 2D gradient $\nabla\mathrm{C}_k$ and (ii) Kullback-Leibler (K-L) divergence $d_{\text{K-L}}(\cdot||\cdot)$ [130]. The former is calculated as

$$\nabla\mathrm{C}_k=\frac{\partial\mathrm{C}_k}{\partial x}\hat{x}+\frac{\partial\mathrm{C}_k}{\partial y}\hat{y}, \tag{5.6}$$

where $\hat{x}$ and $\hat{y}$ are unit vectors defining x and y directions, respectively. Equation 5.6 enables directional alignment of the local vectors of $\nabla\mathrm{C}_k$ for each target to the normal vectors of the perimeter of S_k (Fig. 5.3). The latter enables us to find the orientation within Θ that minimises the divergence between the local intensity distribution of C_k at iteration $h+1$ and the rotated version of $\mathcal{P}(\mathbf{z}_{m,k}^{1,h+1}(\theta))$. Specifically, we have

$$p(\mathrm{C}_k|\mathbf{z}_{m,k}^{1,h+1})=\max_{\theta\in\Theta}\left[p(\mathrm{C}_k|\mathbf{z}_{m,k}^{1,h+1}(\theta))\right]=$$
$$\max_{\theta\in\Theta}\left[\exp\left\{-\frac{1}{2}\left(\frac{E(\nabla\mathcal{C}(\mathbf{z}_{m,k}^{1,h+1}(\theta)),\vec{\mathcal{E}}(\theta))}{\sigma_{\mathcal{C}}}\right)^2\right\}\cdot\exp\left\{-\frac{1}{2}\left(\frac{d_{\text{K-L}}(\mathcal{N}(\boldsymbol{\mu}_L,\boldsymbol{\Sigma}_L)||\mathcal{P}(\mathbf{z}_{m,k}^{1,h+1}(\theta)))}{\sigma_{\text{K-L}}}\right)^2\right\}\right], \tag{5.7}$$

where, with a simplified notation, the argument θ indicates the rotated version of the state. $\nabla\mathcal{C}(\mathbf{z}_{m,k}^{1,h+1}(\theta))$ is the 2D gradient of C_k corresponding to the pixels adjacent to the perimeter $\mathcal{E}(\theta)$, $\vec{\mathcal{E}}(\theta)$ are the normal vectors of the θ-rotated ellipse perimeter (Fig. 5.3), and $\sigma_{\mathcal{C}}$ and $\sigma_{\text{K-L}}$ are constants. $\boldsymbol{\mu}_L$ and $\boldsymbol{\Sigma}_L$ are the components obtained by fitting a 2D Gaussian [3] in the domain $r_1\times r_1$ of $\mathbf{z}_{m,1}^{1,h+1}$ on C_k. $E(\cdot)$ is a function that quantifies the orientation error of the ellipse with respect to the direction of the gradient. The goal is to minimise the error between $\nabla\mathcal{C}(\mathbf{z}_{m,k}^{1,h+1}(\theta))$



(a) (b)

Figure 5.3: Example of detection alignment using the error between the vectors of the gradient on the ellipse perimeter (black vectors) and normal vectors to the ellipse perimeter (white vectors). The goal is to minimise the error between the two sets of vectors: (a) case with a larger error due to misalignment; (b) case with a smaller error.

and $\vec{\mathcal{E}}(\theta)$:

$$E(\nabla\mathcal{C}(\mathbf{z}_{m,k}^{1,h+1}(\theta)),\vec{\mathcal{E}}(\theta)) = ||\nabla\mathcal{C}(\mathbf{z}_{m,k}^{1,h+1}(\theta)) - \vec{\mathcal{E}}(\theta)||_2, \tag{5.8}$$

where $||\cdot||_2$ is the ℓ-2 norm. When all the iterations are performed by M-H, multiple detections may converge to the same target. As last step, in order to suppress these detections, NMS is again applied but on the converged detections using τ_{nms} on the overlap to obtain $\mathcal{Z}_k^1$ (Fig. 5.2(g,h)).

5.2.2 Detector based on hierarchical-isocontour and morphology

Targets can appear at different intensity levels within the same frame (e.g. due to illumination changes) and a single intensity level may not be enough to separately detect all the targets. In fact, the gradient-climbing based detector can be inaccurate in the case of multi-peak intensity over the target (see detections in region 1 of Fig. 5.2(g) and Fig. 5.2(m)). Therefore, in order to distinguish adjacent targets with different intensity levels, we exploit the intensities at which a target appears as a single object. Since it is generally challenging to distinguish them at low-intensity levels (see adjacent targets in the middle of Fig. 5.2(k)), we detect targets by "slicing" the intensity map at different intensity levels (isocontours). Each level is then used to infer size and shape properties of the target. Let $\mathcal{Z}_k^2 = \{\mathbf{z}_{m,k}^2\}_{m=1}^{M_k^2}$ be the subset of detections $\mathbf{z}_{m,k}^2$ that is finally inferred with this method, where M_k^2 is the number of detections at k. $\mathbf{z}_{m,k}^2$ has the same elements of (Eq. 5.1).

Let $\mathcal{I}_{\tau_{iso},k} = g_{\tau_{iso}}(\mathrm{C}_k)$, with $\tau_{iso} \in [0,1]$, be the isocontours extracted from the target-intensity map C_k at layer τ_{iso}, where the function $g_{\tau_{iso}}(\cdot)$ computes the isocontours [91] on C_k. Values of

τ_{iso} close to 0 might provide large regions encapsulating multiple adjacent targets, whereas values of τ_{iso} close to 1 might provide small regions with high intensity values, with the chance of discarding targets with low intensity values. In order to detect the targets appearing at different intensity levels, isocontours are computed by ranging τ_{iso} in the interval Ω (multiple layers) and the discretisation of the values within Ω can be manually chosen. To separate regions connected by thin segments and to filter out background clutter, each layer $\mathcal{I}_{\tau_{iso},k}$ is processed with morphological operators, which include hole filling [158] followed by erosion and dilation [58]. At each τ_{iso}, we select the connected regions taking into account shape information [120]. We use eccentricity for the elliptic model. In order to select reliable target regions, we choose an eccentricity of 0.75. The selected regions are then used to determine the detections of the initial set $\tilde{\mathcal{Z}}_k^2$. Each $\tilde{\mathbf{z}}_{m,k}^2$ has the same elements as those in (Eq. 5.1) and their values are defined according to the following properties: $(\tilde{x}_{m,k}^2, \tilde{y}_{m,k}^2)$ is defined by the region centroid, $\tilde{\theta}_{m,k}^2$ is the region orientation, $\tilde{\imath}_{m,k}^2$ is initialised at zero value and r_1, r_2 are defined a priori.

Extracting regions at multiple layers of isocountours may lead to multiple spatially-close detections for each target. Hence, we cluster detections in order to remove redundant information. We use Mean-Shift (MS) [39] to cluster neighbouring detections by using the position information of the detection of $\tilde{\mathcal{Z}}_k^2$, without any prior knowledge on the number of clusters and with a fixed kernel size.

Let the kernel size be r_2 and the set of clusters $\Psi_k = \{\psi_{r,k}\}_{r=1}^{\mathcal{R}_k}$, with $\psi_{r,k}$ the generic r^{th} cluster and $\mathcal{R}_k$ the set of cluster indexes. For each $\psi_{r,k}$, we generate a detection $\mathbf{z}_{m,k}^2$ whose position coincides with the centroid position $(x_{m,k}^2, y_{m,k}^2)$ of the cluster. The orientation $\theta_{m,k}^2$ is calculated as the circular median [22] of the states belonging to the cluster and $\tilde{\imath}_{m,k}^2$ is calculated as defined in Sec. 5.2 within the area defined by the ellipse $(r_1, r_2, \theta_{m,k}^2)$ centred in $(x_{m,k}^2, y_{m,k}^2)$.

Fig. 5.2(i,k,m) and Fig. 5.2(j,l,n) show examples of the method on bee and ant datasets, respectively. Fig. 5.2(i,j) show the levels of isocountour applied to the target-intensity maps. Fig. 5.2(k,l) shows results after applying morphology to the isocontours with $\tau_{iso} = 0.7$ and 0.6, respectively. The respective final outputs of hierarchical-isocontour based morphology before MS clustering is shown in Fig. 5.2(m,j).

5.2.3 Pruning and fusion of candidate detections

After merging the detections of gradient-climbing based and hierachical-isocontour based detectors, $\check{\mathcal{Z}}_k = \mathcal{Z}_k^1 \cup \mathcal{Z}_k^2 = \{\check{\mathbf{z}}_{b,k}\}_{b=1}^{\check{B}_k}$, where each $\check{\mathbf{z}}_{b,k}$ of $\check{\mathcal{Z}}_k$ is obtained by the intersection operation,

we aim to eliminate the remaining false positives and repeated detections of the two methods. As done in Sec. 5.2.2, we cluster neighbouring detections of $\check{\mathcal{Z}}_k$ using MS with a kernel of size equal to the minor semi-axis r_2. Let $\check{\Psi}_k = \{\check{\psi}_{r,k}\}_{r=1}^{\check{\mathcal{R}}_k}$ be the resulting set of clusters, where $\check{\mathcal{R}}_k$ is the number of clusters at frame k. For each cluster $\check{\psi}_{r,k}$, a single detection $\mathbf{z}_{b,k}$ is selected with the highest $\check{\imath}_k$, such that

$$b = \underset{b^* \in \check{\psi}_{r,k}}{\arg\max}(\check{\imath}_{b*,k}), \tag{5.9}$$

where $\check{\imath}_{b^*,k}$ is the $\check{\imath}_k$ term defined within each $\check{\psi}_{r,k}$. $\mathcal{Z}_k$ will therefore be composed of the detections $\check{\mathbf{z}}_{b,k}$ with largest $\check{\imath}_{b^*,k}$ from each cluster.

The block diagram depicting the main steps of the target detector is shown in Fig. 5.1.

5.3 Graph-based association

The association process enables the linkage of detections over time, the generation of detections in frames with miss-detections, and the pruning of isolated and false detections. We use only target-position information as a feature for the generation of tracks. In order to reduce the complexity of the association process, we initially generate and subsequently link short tracks [73, 189].

Let the set of initial tracks, $\mathfrak{T} = \{\mathfrak{t}_l\}_{l=1}^{L}$, be generated by a tracking algorithm such as MT-TBD or a different algorithm that performs a sequential association of detections. In the latter case, a widely used method optimally associates consecutive detections using the Hungarian algorithm (HA)[1] [36, 73, 189]. Short tracks are generated by enforcing high similarity between detection in order to ensure that the initial association is performed without errors while reducing the complexity of the overall problem. The sequential association is performed while keeping unique identities to the associated detections. Thus for each pair $(\mathcal{Z}_k, \mathcal{Z}_{k+1})$ we calculate the cost $\mathfrak{C}_{k,k+1} \in \mathbb{R}^{B_k \times B_{k+1}}$ using the ℓ-2 norm between each position state in frame k and $k+1$, $\mathfrak{c}_{k,k+1}^{n,n'} = ||(x_{n,k}, y_{n,k})^T - (x_{n',k+1}, y_{n',k+1})^T||_2$, where $\mathfrak{c}_{k,k+1}^{n,n'}$ is the element of $\mathfrak{C}_{k,k+1}$ on the row n and column n'.

Longer tracks $\mathcal{T} = \{\mathsf{T}_a\}_{a=1}^{A}$ are generated by sequentially linking short tracks as a MAP problem [189],

$$\mathcal{T}^* = \underset{\mathcal{T}}{\arg\max}\, p(\mathcal{T}|\mathfrak{T}), \tag{5.10}$$

[1] http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html, last accessed: December 2013.

with $\mathcal{T}^*$ the set of tracks with the highest probability. The direct maximisation of (Eq. 5.10) is computationally expensive because the number of combinations of the elements of the set $\mathfrak{T}$ is large [73]. Methods exploiting dynamic programming, such as the Viterbi algorithm [178], can find the global optimal solution that maximises the problem by dividing the overall problem into simpler subproblems. The Viterbi algorithm for multi-target tracking assumes that all the nodes (in our case the short tracks) of a graph should be linked to each other and the links should be unmerged. Since the graph we are going to built may contain many false positive nodes, the use of the Viterbi algorithm would lead to the association of nodes that should not be associated. An example can be when all the good nodes are connected and only false positive nodes are left. The Viterbi algorithm would also connect these false positive nodes which would result in an increase in false positive tracks. Therefore, we propose a greedy graph-based (GGB) method that enables the linkage of short tracks by introducing latency and by discarding false positives. We decompose the problem as

$$\begin{aligned} p(\mathcal{T}|\mathfrak{T}) = p(\mathsf{T}_1|\mathfrak{T})\cdot \\ p(\mathsf{T}_2|\mathfrak{T} \setminus \mathsf{T}_1)\cdot \\ p(\mathsf{T}_3|\mathfrak{T} \setminus \mathsf{T}_2, \mathsf{T}_1)\cdot \\ \dots \\ p(\mathsf{T}_A|\mathfrak{T} \setminus \mathsf{T}_{A-1}, ..., \mathsf{T}_a, ..., \mathsf{T}_1), \end{aligned} \tag{5.11}$$

and we maximise each probability term iteratively with a greedy process. This enables us to perform tracking within a short temporal buffer and, unlike [155] or [73], once a trajectory is computed within the buffer, we do not change the solution afterwards. The advantage of this formulation involves the possibility of employing the tracking algorithm for time-critical applications, where online trajectory solutions (with a short delay) are needed. On the other hand, the main disadvantage involves the impossibility of refining tracking solutions once the association is performed within the shifting temporal window.

Let $G = (E, \mathfrak{T})$ be a graph, where E is the set of edges whose weights are calculated via a link probability and $\mathfrak{T}$ are the nodes. Each node is composed of a sink (child) and a source (parent), denoted as $\mathfrak{t}_l^-$ and $\mathfrak{t}_l^+$, respectively. We define a function $g(\cdot)$ that links short tracks by performing a non-linear interpolation of the positions in order to generate detections among

Algorithm 1 *Greedy graph-based association*

$\mathfrak{T}$: set of temporally-ordered short tracks. $\ell(\mathfrak{t}_{l'}|\mathfrak{t}_l)$: link probability.
τ_ℓ: threshold for negligible link probabilities. $(\mathfrak{B},\mathfrak{b})$: buffer size and temporal shift.
$\mathfrak{T}_{proc}$: processed short tracks. $L_{\mathfrak{B}}$: number of short tracks within the buffer $\mathfrak{B}$.

$\mathcal{T} \leftarrow \emptyset$; $\mathfrak{T}_{proc} \leftarrow \emptyset$
for $l \leftarrow 1$ to $L_{\mathfrak{B}}$ **do**
  $\mathcal{T}_{temp} \leftarrow \emptyset$; $\mathfrak{t}_l \leftarrow \mathfrak{T}$
  if $\mathfrak{t}_l \notin \mathfrak{T}_{proc}$ **then**
    $\mathcal{T}_{temp} \leftarrow \mathfrak{t}_l$
    while (1) **do**
      $\mathfrak{T}_{\tau_\ell} \leftarrow$ findnodes *s.t.* $\{\ell(\mathfrak{t}_f^-|\mathfrak{t}_l^+) > \tau_\ell, \mathfrak{t}_f \notin \mathfrak{T}_{proc}, \mathfrak{t}_f \in \mathfrak{T}, f > l\}$
      if $\mathfrak{T}_{\tau_\ell} \neq \emptyset$ **then**
        while 1 **do**
          $\mathfrak{t}_{l'} = \arg\max_{\mathfrak{t}_f \in \mathfrak{T}_{\tau_\ell}} \ell(\mathfrak{t}_f^-|\mathfrak{t}_l^+)$
          if $(\arg\max_{\mathfrak{t}_f \in \mathfrak{T}_{\tau_\ell}} \ell(\mathfrak{t}_f^+|\mathfrak{t}_l^-) = \mathfrak{t}_l) || (\mathfrak{t}_{l'} = \emptyset])$ **then**
            $\mathcal{T}_{temp} \leftarrow \mathfrak{t}_{l'}$; $\mathfrak{t}_l \leftarrow \mathfrak{t}_{l'}$
            break while
          else
            $\mathfrak{T}_{\tau_\ell} \leftarrow \mathfrak{T}_{\tau_\ell} \setminus \mathfrak{t}_{l'}$
          end if
        end while
      else
        break while
      end if
    end while
    $\mathfrak{T}_{proc} \leftarrow \mathcal{T}_{temp}$; $\mathcal{T} \leftarrow g(\mathcal{T}_{temp})$
  else
    $\mathfrak{T}_{proc} \leftarrow \mathfrak{t}_l$; $\mathcal{T} \leftarrow g(\mathfrak{t}_l)$
  end if
end for

linked short tracks and to smooth tracks $\mathcal{T}$. Equation (5.11) can be solved by formulating the problem with a graph and using the concept of *parents* and *children*. A *parent* is a short track that ends before the start of another one, which in turn is defined as a *child*. A parent can be associated with a child when the likelihood (weight) from the parent to the child is the biggest for the parent and also the biggest for the child with respect to other competitive (or candidate) parents. We aim to associate parents and children with a forward association and a backward validation, in order to achieve the best association between two short tracks with respect to the competing candidates. Hence, we iteratively and pair-wisely match parents and children over time until there are no alternative pairings in which the single best match is found between each parent and child.

Equation 5.11 assumes that each track is temporally dependent on another track conditioned upon their initial state: $s_l \leq s_{l+1} \leq \ldots \leq s_L$, where s_l denotes the frame of the initial state of a generic $\mathfrak{t}_l$ and likewise for e_l denoting the frame of the last state. In fact, the calculation of T_a depends on $\mathfrak{T}$, except those used for the calculation of T_{a-1}, which in turn depends on $\mathfrak{T}$, except those used for T_{a-2}, and so on. Each probability term of (5.11) is maximised via a recursive

process using a link probability $\ell_1(\cdot)$ between short-track pairs $(\mathfrak{t}_l, \mathfrak{t}_{l'})$, with $l \neq l'$,

$$\ell_1(\mathfrak{t}_{l'}|\mathfrak{t}_l) = \begin{cases} \exp -\frac{1}{2}\left\{\left(\frac{\beta_{l',l}}{2\sigma_1}\right)^2 + \frac{(s_{l'}-e_l)^2}{2\sigma_2}\right\} & \text{if } e_l - s_{l'} < \tau_l \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

where τ_l is the temporal interval permitted between the end of a short track and the start of another, $\beta_{l',l}$ is a normalised ℓ-2 norm

$$\beta_{l',l} = \frac{1}{r_1}||(x_{l,e_l}, y_{l,e_b})^T - (x_{l',s_{l'}}, y_{l',s_{l'}})^T||_2, \tag{5.13}$$

and σ_1, σ_2 are constants. We use a normalised distance over the shape model, so that σ_1 is target-size independent.

The linking process is performed within a buffer of duration $\mathfrak{B}$ frames, implemented as a sliding window approach with $\mathfrak{b}$ overlapping frames and with $\tau_l > 0$. The linkage method is described in Algorithm 1.

Temporally-overlapping tracks for short periods of time can occur when concurrent and multiple detections are generated for a single target. This is a well known problem that is usually addressed with NMS at the detection stage [51]. Sometimes NMS can fail if the overlap of the detections is below the threshold set in the algorithm. Hence, by employing this additional analysis in GGB, we can reduce the risk of track fragmentation[2] due to concurrent detection on the same target. Hence, after having generated the set $\mathcal{T}$, we reapply Algorithm 1 using $\mathfrak{T} = \mathcal{T}$ and the likelihood function

$$\ell_2(\mathfrak{t}_{l'}|\mathfrak{t}_l) = \begin{cases} \exp -\frac{1}{2}\left\{\left(\frac{\bar{\beta}_{l',l}}{2\sigma_1}\right)^2\right\} & \text{if } \tau_o \leq s_{l'} - e_l \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.14}$$

where $\tau_o \leq 0$ is the number of overlapping frames and

$$\bar{\beta}_{l',l} = \frac{1}{|s_{l'} - e_l|}\frac{1}{r_1} \cdot \sum_{\rho=1}^{|s_{l'}-e_l|-1} ||(x_l(e_l-\rho), y_l(e_l-\rho))^T - (x_{l'}(s_{l'}+\rho), y_{l'}(s_{l'}+\rho))^T||_2. \tag{5.15}$$

Finally, we prune tracks within the buffer $\mathfrak{B}$ with a shorter duration than a threshold τ_D.

[2]It occurs when the track of a same target terminates in a frame and restarts with another identity after few frames.

5.4 Results

We evaluate the performance of the proposed detection approach, which is the combination (Fusion) of the gradient-climbing based detector (GCD) and hierarchical-isocontour based morphology (HIM). We then evaluate the performance of the greedy graph-based association (GGB). We also assess the sensitivities of detection and tracking by changing the parameters, such as the ellipse size, isocontour thresholds, buffer size, likelihood standard deviations and temporal gap permitted to merge short tracks.

5.4.1 Methods for comparison

We compare the detector with four detection approaches (D) and the trackers with four alternative trackers (T). D1: threshold based plus Mean-Shift clustering applied on C_k (similar to [84]). D2: D1 followed by Non-Maxima Suppression (NMS). D3: template matching on grayscale frames via normalised cross-correlation [58] using eight target patches at different orientations cropped from the videos. D4: D3 followed by NMS. D5: maximally stable extremal regions (MSER) [115] with MS clustering and NMS. T1: a baseline hierarchical detection association [1] where the detections are associated frame-by-frame with the Hungarian algorithm in order to generate short tracks, which are further globally associated using the nearest neighbour algorithm in order to generate longer tracks. T2: a multi-particle tracker that employes the Brownian motion as prior knowledge on the target motion [42]. The detection association is done by maximising the probability of finding each target from one frame to the next. T3: based on multiple Kalman filters used to predict and update the locations of the targets [64]. The prediction is performed with a linear motion model, and the association between detections and trackers is performed with the Munkres algorithm. T4: formulated as an energy minimisation problem between piecewise polynomials (B-splines) and target trajectories [15]. We compare T1, T2, T3 and T4 with combinations of the proposed trackers: MT-TBD, MT-TBD+postprocessing (MT-TB+PP), MT-TBD+GGB, HA and HA+GGB. MT-TDB+PP is used with a buffer of 50 frames, as that of the GGB.

5.4.2 Experimental setup

We use a bee dataset (B-D) and an ant dataset (A-D). B-D is composed of 28400 frames of size 640×350 and recorded at 29.97 frame-per-second (fps). We use two clips of video footage

extracted from B-D to quantitatively evaluate the detector and the tracker[3], namely B-D1 (frames 500 to 999) and B-D2 (frames 25500 to 25999): the former contains a higher density of targets compared to the latter. A-D is composed of 10400 frames of size 720×480 recorded at 29.97 fps and we use the whole sequence for the quantitative evaluation.

In B-D, the target-intensity map is the equalised red channel (RGB colorspace) of the frames with a Gaussian filter applied to it. In our experiments, the red channel is found to be a reliable feature since most of the information about the colour of bees lies on this channel. By observing the size of the objects on the image plane we set the ellipse prior size to $r_1 = 42$, $r_2 = 18$. The threshold used for NMS is $\tau_{nms} = 0.3$. The number of iterations for MCMC is $\mathcal{H} = 10$ and the likelihood function used for its computation has variance parameters set as $\sigma_{\mathcal{C}} = 1$ and $\sigma_{\text{K-L}} = 0.7$; these iteration values and standard deviation provide an accurate shape alignment. Smaller values of $\sigma_{\mathcal{C}}, \sigma_{\text{K-L}}$ and larger number of iterations do not further improve the fitting accuracy, whereas larger values of $\sigma_{\mathcal{C}}, \sigma_{\text{K-L}}$ and smaller number of iterations might provide less accurate alignments. τ_{iso} values range in the interval $\Omega = [0.5\ 0.8]$ with a step size of 0.05. The link probability of (Eq. 5.12) has $\sigma_1 = 0.3$ to penalise detections outside the area of the target, and $\sigma_2 = 10$, to link detections for short temporal gaps. The buffer is $\mathfrak{B} = 50$ frames with a temporal shift $\mathfrak{b} = 5$ frames, $\tau_l = 10$ $\tau_o = -10$ and $\tau_D = 15$. In A-D the target-intensity map is the grey-level image and is filtered with a Gaussian function. The parameters are the same as for B-D apart from the ellipse size prior, $r_1 = 16$, $r_2 = 7$, and $\tau_D = 30$. A sensitivity analysis of the detector and tracker parameters is performed later in the section.

5.4.3 Evaluation measures

We evaluate the performance of the detection and tracking methods in terms of Precision (P), Recall (R) and robustness to ID switches. P and R are calculated as defined in Sec. 2.5. A true positive happens when the distance between the estimated location of a target and its ground truth is smaller than a threshold τ_{TP}. In our experiments we use $\tau_{TP} = 30$ pixels for B-D and $\tau_{TP} = 10$ pixels for A-D.

We introduce a new measure to quantify the robustness of a tracker to ID switches by using a two-element vector measure IDSR = $[\Gamma\ \Lambda]$ (ID Switch Rates). IDSR enables us to measure the

[3] Video results of the full sequence can be found here: ftp://motinas.elec.qmul.ac.uk/pub/bee_results/bee_28400.zip

Table 5.1: Detection results. The threshold on the distance used to define a detection result as a true positive is 30 pixels for B-D1 and B-D2, and 10 pixels for A-D. Key: D: Dataset; P: Precision; R: Recall. F: F-Score. D1-D5: alternative detectors.

Target detector	B-D1			B-D2			A-D		
	P	R	F	P	R	F	P	R	F
D1 ([84]+MS)	.63	.90	.74	.61	.81	.70	.60	.93	.73
D2 (D1+NMS)	.80	.71	.75	.76	.66	.71	.91	.88	.89
D3 ([58])	.63	.59	.61	.70	.64	.67	.89	.78	.83
D4 (D3+NMS)	.73	.52	.61	.82	.57	.67	.91	.77	.83
D5 ([115]+MS+NMS)	.73	.81	.77	.79	.84	.83	.98	.91	.94
GCD	.80	.89	.84	.86	.90	.88	.98	.97	.98
HIM	.92	.73	.81	.96	.71	.82	.98	.93	.95
Fusion	.81	.88	.84	.89	.89	.89	.99	.97	.98

ID switches per frame, Γ, and ID switches per track, Λ. Γ is defined as

$$\Gamma = \frac{IDS}{K}, \tag{5.16}$$

where IDS is the total number of ID switches that occurred in the sequence and K is the total number of frames (see Sec. 5.3). Λ is defined as

$$\Lambda = \sum_{k=1}^{K} \frac{i_k}{\zeta_k}, \tag{5.17}$$

where i_k is the number of ID switches and ζ_k is the maximum number of ID switches that can occur at frame k. A small value of τ_{TP} is more suitable to correctly evaluate IDSR. A large τ_{TP} may lead to errors in the evaluation procedure when the optimal association between ground-truth and estimated tracks is computed.

5.4.4 Target detection

We compare the results obtained with the proposed detector and those obtained with alternative approaches (Table 5.1).

Quantitative results

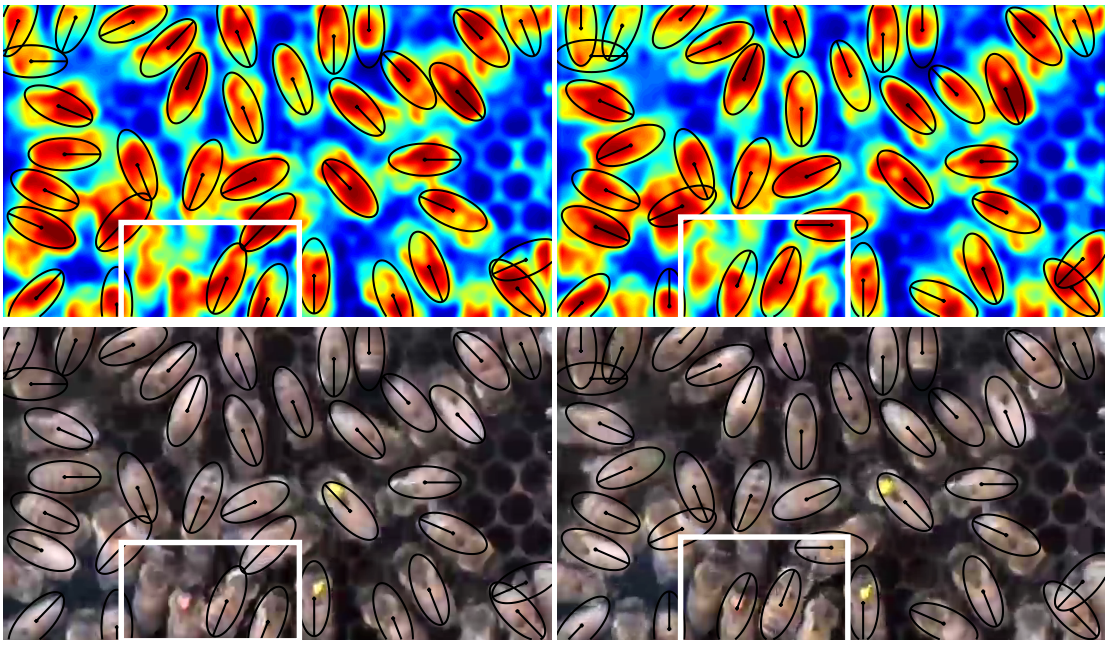
Table 5.1 shows that on average the gradient-climbing based detector (GCD) has higher Recall (R) than the other methods. Hierarchical-isocontour based morphology (HIM) provides the highest Precision (P) compared to the other methods. The fusion operation provides the highest F. By fusing the results we can improve P of GCD, since some of the false positives are discarded by (Eq. 5.9). This fact is visible on B-D1 and B-D2, but is not clear on A-D since the overall values are already high. Morphological operators can be very accurate because they can effectively

filter out clutter, but they might not be able to provide reliable detections in the case of adjacent targets. Even if D1 and D2 provide reasonable results, e.g. D2 reaches F=0.89, which does not contain as challenging situations as B-D, their performance is still lower than those provided by the proposed method. Interestingly, NMS on D2 effectively prunes spurious detections in A-D, but not in B-D1 and B-D2 because NMS suppresses valid detections in the case of adjacent targets when their detected areas overlap. Overall, template-based approaches (D3 and D4) have the lowest performance compared to the other methods. This is because in the case of a high density of targets, template matching is not reliable due to its inability to discriminate adjacent targets, especially when dealing with low-SNR sequences (B-D). The same problem occurs in the case of D5, where MSER features are unable to generate separate regions when targets are close to each other.

Finally, we employ the detector proposed in [84] (only for A-D) followed by Mean-Shift clustering in order to obtain a single detection for each target. This method provides good P=0.98 and R=0.94, but the performance remains lower than that of the proposed method, which reaches P=0.99 and R=0.97. Overlapping and adjacent targets are, in fact, difficult to separate with the method proposed in [84].

Qualitative results

Fig. 5.4 shows sample results containing targets detected with correct orientations. There can be some challenging situations, such as the miss-detected targets in the white bounding box. Firstly, the top-left one has a low intensity compared to the others and it is likely that the detection is converged to one of those neighbours. Secondly, for the target at the bottom of the white box, the detection cannot converge and align to it as it is partially occluded by the neighbouring target and partially outside the frame. Moreover, the morphological operations struggle to detect the target since its intensity values are connected with those of the right-hand-side target. When this last target moves upwards and the neighbouring one moves slightly farther away, it gets detected (Fig. 5.4(b)). In both Fig. 5.4(a) and (b) there is a target (under the top-left corner of the white box) with intensity values more spread out than the others (a bee with open wings) and the orientation of the state incorrectly matches with that of the real target. However, the estimated orientation is aligned to the distribution of the highest intensity values. Indeed, the error in the detection estimation is caused by the fact that we are not using any complex prior knowledge about the targets, unlike [127], other than that they are approximated as elliptical shapes.



(a) (b)

Figure 5.4: Sample detections on low-SNR images (B-D) superimposed on the target-intensity maps and on the respective original frames. Most of the targets are correctly detected on the low-SNR images: (a) a failure of detected targets can be spotted in the white box; this is due to its weak intensity with respect to the other targets; the target on the bottom is not detected because it is partially outside the frame. (b) When the target becomes more visible it gets detected.

Sensitivity

The sensitivity of the detector is assessed by changing the size parameters of the ellipse (r_1, r_2). The experimentation is performed on B-D1 and B-D2 since these are more challenging sequences than A-D as they contain a higher density of targets performing sudden motion variations. Fig. 5.5(a) shows that smaller (r_1, r_2) leads to higher R and lower P. This is due to the multiple detections that are converged on single targets, and since they are not accurately aligned to them, they are not pruned by NMS. While increasing the values of (r_1, r_2), R does not decrease as fast as the increase of P, meaning that the fitting process aligns the shape while enabling an effective pruning with NMS. Small variations of the size, between $r_1 = [39\ 44]$ and $r_2 = [17\ 21]$ do not affect the performance considerably. Interestingly, R in B-D2 increases faster than that in B-D1 due to the lower density of targets, as the detector is less biased by intensity values of adjacent targets. P follows a similar trend for both cases.

Moreover, we assessed the sensitivity of HIM on B-D1 for different values of Ω and τ_{iso} (Fig. 5.5(b,c)). P is the highest for τ_{iso}>0.5, whereas R remains at low levels (below 0.60) throughout all the variations of τ_{iso}. R is greater than zero for τ_{iso}>0.4 and τ_{iso}<0.9, and it

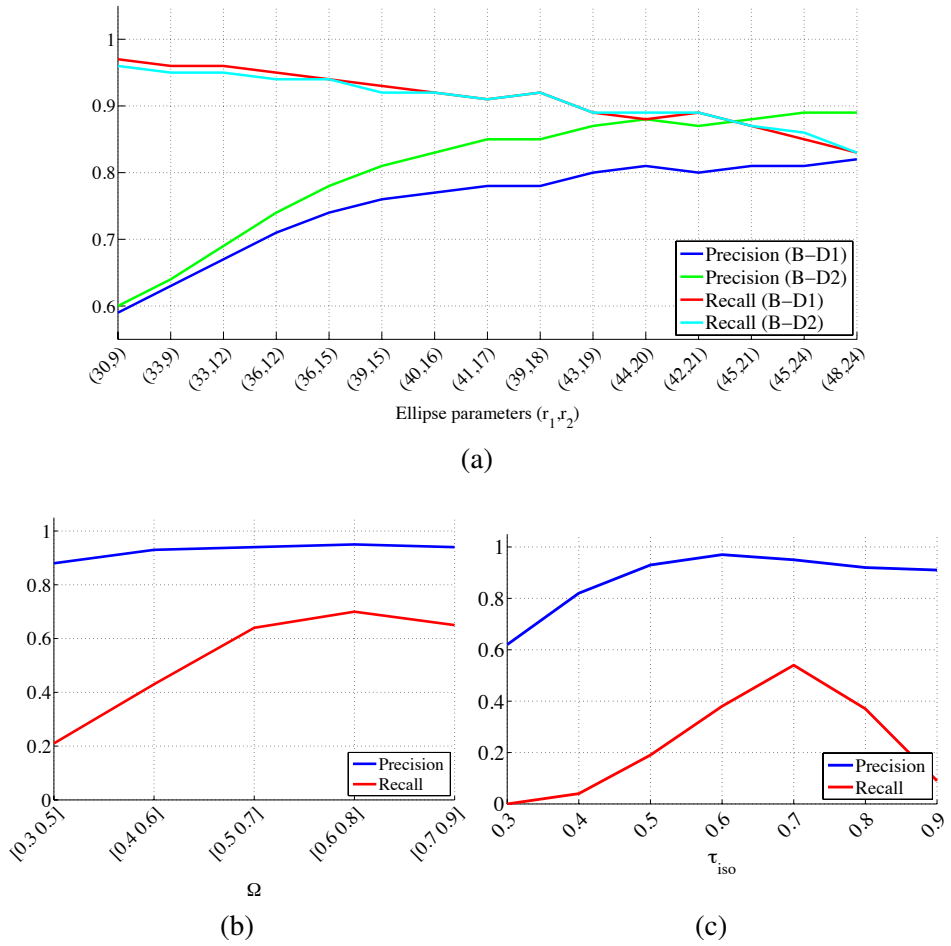


Figure 5.5: Sensitivity of the proposed detector (Fusion) by changing (a) shape parameters (r_1, r_2) on B-D1 and B-D2, and isocontour-based approach by changing values of (b) the interval Ω and (c) applying single values of τ_{iso}.

reaches the highest value (0.54) for τ_{iso}=0.7. This is the reason we employed a multilayer-isocontour approach with Ω=[0.5 0.8]. Indeed, values outside this range (e.g. [0.3 0.9]) do not further improve the performance. In particular, for τ_{iso}<0.5 HIM would mainly outline clutter and big regions would be discarded by the shape constraint (eccentricity). On the other hand, for τ_{iso}>0.8 isocontours would outline small and negligible regions.

5.4.5 Target tracking

We analyse the performance of GGB and compare it with alternative trackers (Sec. 5.4.1). The trackers are tested on bee (B-D) and ant (A-D) sequences using detections generated by the proposed detector.

Table 5.2: Tracking results. The threshold on the distance used to consider a tracking result as a true positive is 30 pixels for B-D1 and B-D2, and 10 pixels for A-D. The results for T4 in A-D are not provided due to implementation limitations with long sequences. Key: D: Dataset; P: Precision; R: Recall; IDSR: ID Switch Rates. T1-T4: alternative trackers.

Trackers	B-D1				B-D2				A-D			
	P	R	F	IDSR	P	R	F	IDSR	P	R	F	IDSR
T1 ([1])	.81	.88	.84	[.60 9.53]	.89	.89	.89	[.43 6.55]	.98	.97	.98	[.09 44.50]
T2 ([42])	.83	.86	.84	[.42 6.67]	.91	.88	.89	[.26 3.92]	.98	.97	.98	[.07 34.65]
T3 ([64])	.59	.93	.72	[.80 12.74]	.81	.93	.87	[.68 10.50]	.95	.98	.96	[.08 42.00]
T4 ([15])	.76	.82	.79	[2.2 35.40]	.87	.90	.88	[1.05 16.16]	-	-	-	-
MT-TBD	.83	.84	.84	[1.35 21.68]	.90	.86	.88	[1.15 17.52]	.96	.98	.97	[.21 109.55]
MT-TBD+PP	.81	.85	.83	[.29 4.70]	.90	.87	.88	[.21 3.18]	.97	.97	.97	[.13 66.50]
HA	.81	.88	.84	[1.97 31.53]	.89	.89	.89	[1.69 25.94]	.98	.97	.98	[.26 135.05]
MT-TBD+GGB	.83	.87	.85	[.22 3.51]	.90	.88	.89	[.17 2.61]	.98	.97	.98	[.06 30.50]
HA+GGB	.82	.89	.85	[.22 3.55]	.90	.91	.91	[.14 2.14]	.99	.98	.99	[.03 14.00]

Bee sequence

Tracking results on B-D1 and B-D2 are shown in Table 5.2, and we can see that overall the proposed method (HA+GGB) outperforms the other methods: on average F is the highest and ID switches are fewer. T3 has the lowest P in both B-D1, B-D2 and A-D, due to the prediction step of the Kalman filter (KF) when no detections are available for the update. Specifically, in situations of abrupt motion changes of the targets, KF is unable to correctly predict the future location, and when no detections are available for the update, the filter has to use the prediction as a valid state (Fig. 5.6(a-c)). Then, KF keeps predicting incorrect states until the error covariance becomes big enough to consider the target lost. Following that, the lost target is re-initialised with a new track. Therefore the tracks generated from spurious predictions increase the false positive rate. T2 has higher P than that of T3 since the prior on the target motion looks closer to the actual movement of the targets (Fig. 5.6(h-j)). However, on average T1 has the same F of T2, but T2 is more accurate at correctly distinguishing target identities. Similarly to T3, MT-TBD uses a linear motion model to predict target locations. However, the update is performed using intensity values instead of detections. This enables more flexibility for the tracker to determine whether detections belong to noise or not; P is in fact higher, while R is lower, which means that correct detections are sometimes considered clutter. The postprocessing applied on MT-TBD (MT-TBD+PP) dramatically improves the performance by getting very close to that of HA+GGB, especially in terms of IDSR. Although MT-TBD shows better performance than HA, when GGB is applied to MT-TBD (MT-TBD+GGB) the performance does not achieve that of HA+GGB. The main reason is the lower R of MT-TBD with respect to that of HA, in fact also MT-TBD+GGB does not achieve an R as good as that of HA+GGB. Finally T4 has very

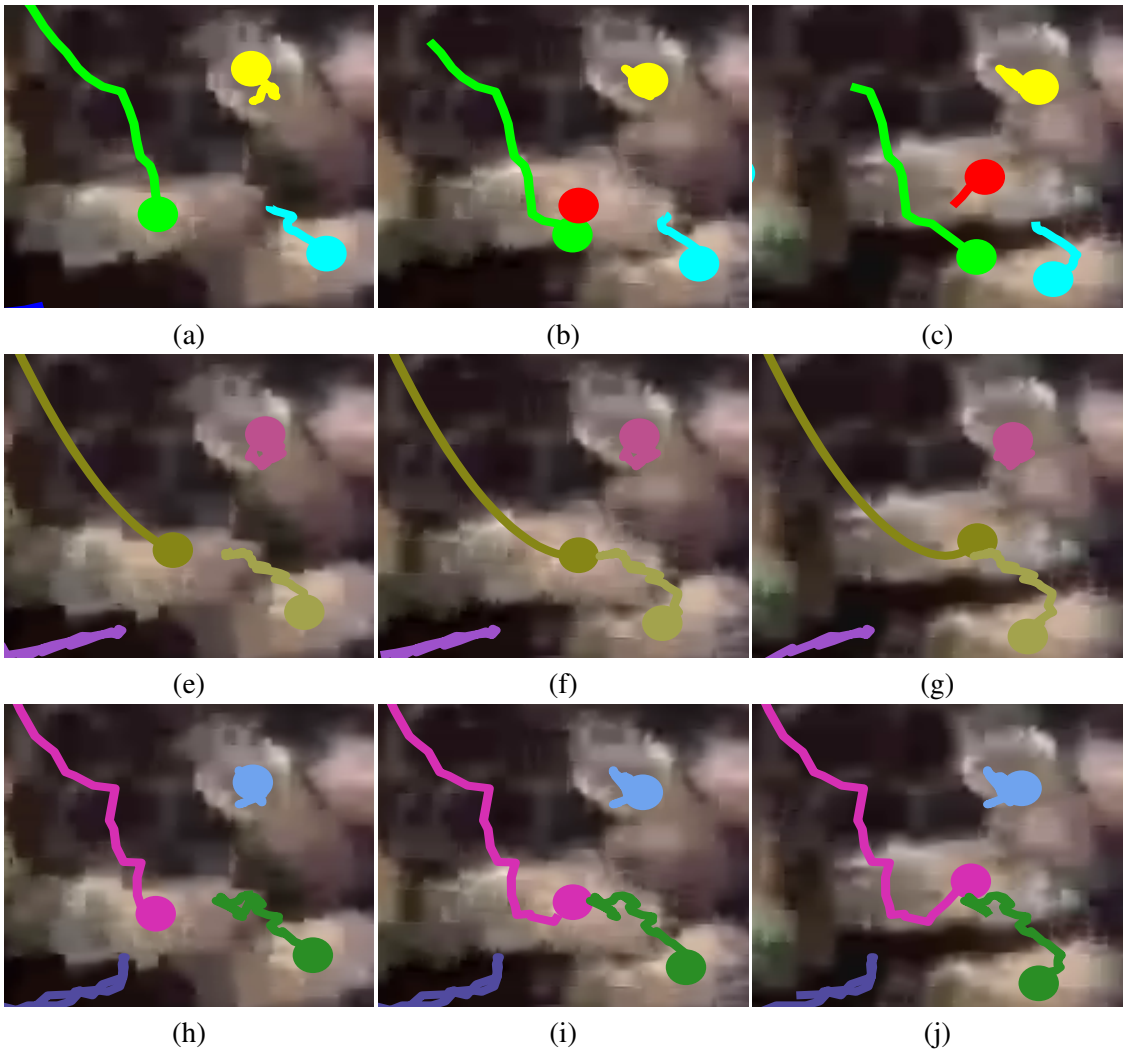


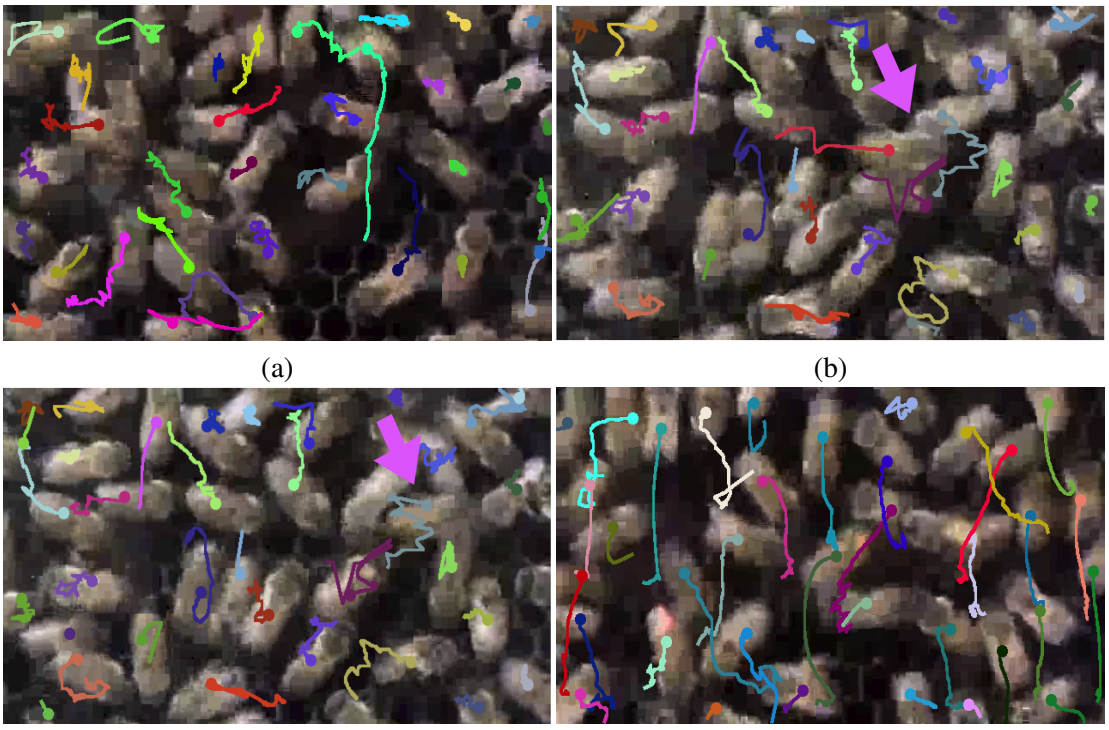
Figure 5.6: Example of an abrupt motion change of a target where (a-c) a Kalman filter-based method can fail. The failure occurs on (a) the green track, where the tracked target is moving from top to bottom and (b) suddenly changes direction. The green track will keep going straight on, while the red track is initialised. (c) The track survives for a few steps before being terminated. (e-g) Multi-particle tracker based on Brownian motion and (h-j) the proposed approach can deal with abrupt motion changes. Different rows and different trajectory colours represent different tracker results.

poor performance compared to the others. From the quantitative results we can observe good performance when the Brownian motion is assumed, but even better performance when a greedy association and simple heuristics (Gaussian models) are employed.

IDSR of the GGB is ten times better than the sequential linking performed with the Hungarian algorithm (HA). Results show that GGB can deal with highly fragmented tracks and reliably associate them while keeping a low ID switch rate. Additionally, GGB, as opposed to PP applied to MT-TBD, provides the best results except in B-D1 where their scores are comparable. Both

GGB and PP account for candidate false-positive short-tracks during the linking and discard them if considered spurious. However, GGB does not use colour as discriminative feature. In situations where a candidate link is likely to be uncertain because spatio-temporal features provide a low weight, the colour feature in PP might increase the weight and lead to an erroneous linking. In fact, the major strength of GGB is its ability to discard false-positive short-tracks during the association process. T2 performs better than T1, T3 and T4. T2 associates track-measurement by maximising the probability (defined by the Brownian diffusion probability) of finding a measurement in the neighbourhood of a track. The ID association of T2 is found to be more reliable than the association with HA (i.e. T1, T3): T1 performs the association of detections iteratively with HA, whereas T3 uses the HA to associate predicted target locations with the measurements. Lastly, T4 does not perform an explicit association, but produces trajectories via the fitting of spline functions that minimise the error in the regression process. Hence a regression method cannot be applied to such a cluttered scenario with targets performing sudden motion variations.

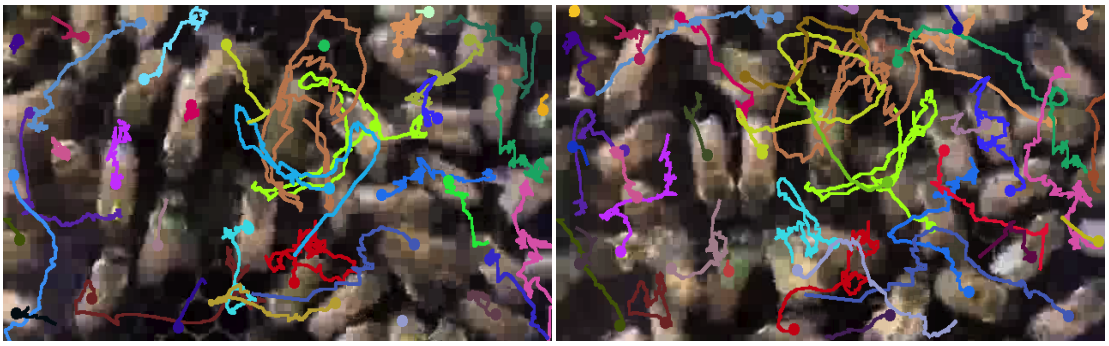
Fig. 5.7 shows targets tracked in highly-populated frames with poor illumination and low resolution (trajectories are truncated at 50 frames to make the visualisation clearer). On the left hand side of Fig. 5.7(a), where there is a high density of targets and the resolution is low, the frame appears as a dark patch and presents artifacts due to image compression. In this region we can spot a few tracking failures that are recovered in the subsequent frames when targets get farther apart (Fig. 5.7(b,c)). These figures also show a case of ID switch between the red identity in the centre of Fig. 5.7(b) and the grey identity on the same path (magenta arrow). The bee with the red identity is moving from left to right and when she passes over the other bee, the grey identity passes to the red one and the red identity gets lost. This is a challenging situation because when the detections of the overlapped bee become available, GGB associates those of the flying bee to those of the still bee. Fig. 5.7(d) shows a case with an abrupt motion change, that is when the camera is moved by the operator. The targets remain tracked and new tracks of targets in the lower part of the image are initialised. In Fig. 5.8 we can notice how the central bee is tracked for more than 200 frames. Similarly to the previous case, the camera is moved by the operator and we can notice it by looking at the position of the dark-orange trajectory in Fig. 5.8(a) and (b). An example of the density of trajectories is shown in Fig. 5.9.



(a) (b)

(c) (d)

Figure 5.7: Sample tracking results on dataset B-D in challenging situations. (a) High density of bees on the left-part of the frame; (b-c) ID switch of the red trajectory in the middle of the frame and grey trajectory on its path (magenta arrow); (d) Robustness of the method to camera movements. The trajectories are truncated to the last 50 frames to make the visualization clearer.

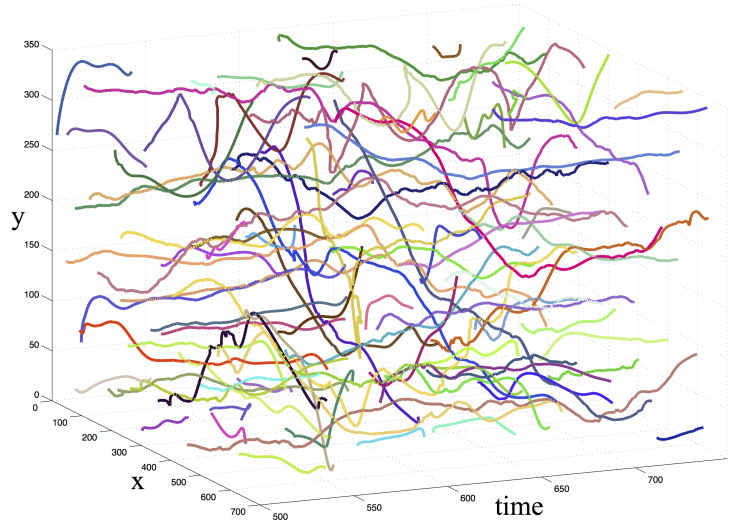


(a) (b)

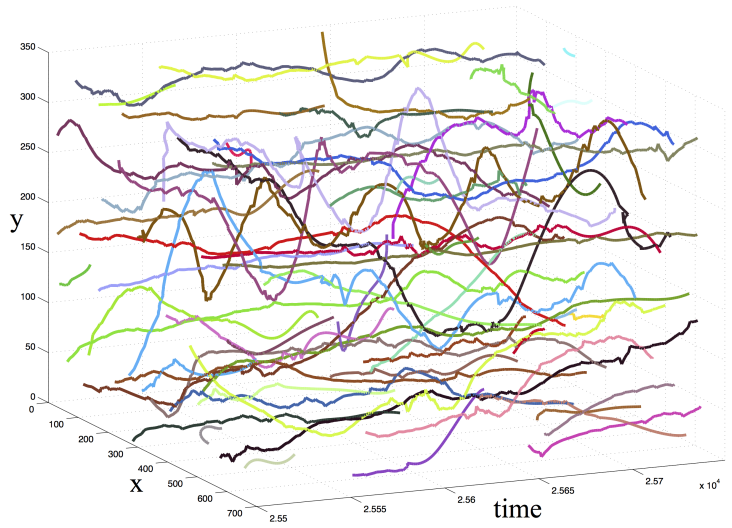
Figure 5.8: Sample tracking results on the bee dataset (B-D) with long term visualisation. Abrupt motion changes are successfully dealt with the proposed tracker. The trajectories from (a) to (b) are all shifted on the top-right because the camera has been moved by the operator. The trajectories are truncated to the last 200 frames to make the visual representation clearer.

Ant sequence

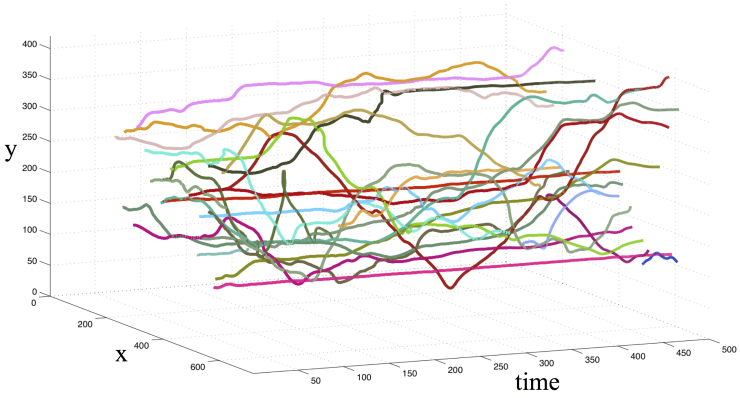
Results of A-D are quantitatively reported in Table 5.2 and qualitative examples are shown in Fig. 5.10. Even if the same sequence is already used in [84] and [174], we cannot compare the results since they employ ground-truth information to initialise the target locations and in the case



(a)



(b)



(c)

Figure 5.9: 3D visualisation of trajectories from (a) B-D1, (b) B-D2 on 250 frames and (c) A-D on 500 frames. Graphs show the trajectories of insects on the image plane (x-y axes) over time.

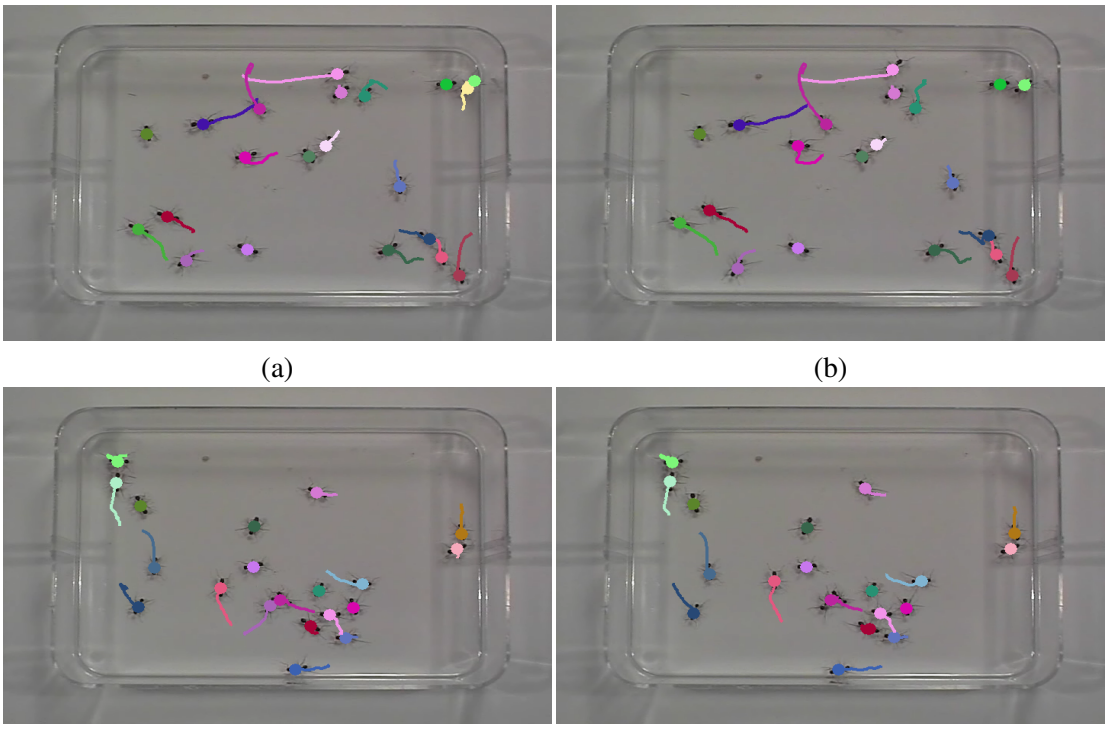


Figure 5.10: Sample tracking results on the ant dataset (A-D). ID switches can be due to (a,b) multiple detections on a single target (top-right corner), or (c,d) crossing/overlapping targets (middle). The trajectories are truncated to the last 200 frames to make the visual representation clearer.

of tracking failures. Whereas, we do not use any manual intervention and we let the tracker run throughout the sequence. The results for T4 in A-D are not provided due to its implementation limitations with long sequences [15].

As we can see from the results, HA+GGB has the highest P and R, and lowest IDSR compared to the other methods. The proposed method has the lowest IDSR and the few ID switches are due to two reasons: first, multiple detections are generated on the same targets when they are close to borders (Fig. 5.10(a,b)) and we can see that it happens because of the reflection of the ant in the glass; second, when targets cross each other (Fig. 5.10(c,d)) there can be track interruptions, which is mainly due to the fact that occlusions have not been explicitly modelled in GGB. Likewise on B-D, T2 has closer performance to HA+GGB, but the quantity of ID switches is still approximately double. Similarly to B-D, the performance of T3 is closer to that of T2 and better than T1. This is due to the different motion of the targets, which can be better approximated with a linear model. Moreover, with the same buffer size as MT-TBD+PP (50 frames), HA+GGB is more robust at keeping the correct identities associated to the targets, even without employing

prior dynamics.

Sensitivity

Similarly to the postprocessing stage presented in Sec. 3.4, we assess the sensitivity of GGB. In particular, we use B-D1 and range (i) the buffer duration $\mathfrak{B}$ in the interval [10 100] frames with step size 10, (ii) the temporal interval permitted to merge short tracks τ_l for values $\{5, 10, 20\}$, and (iii) $\{\sigma_1, \sigma_2\}$ used for the computation of the likelihood function (Eq. 5.12) for values $\{0.15, 5\}$, $\{0.3, 10\}$, $\{0.45, 15\}$. The performance are shown in Fig. 5.11. From the graphs we can observe that the variation of the buffer duration does not largely affect P and R for same values of $\{\sigma_1, \sigma_2\}$. The results show that the tracking algorithm does not largely decrease its performance with short buffer durations (e.g. 10 frames) and has stable performance at large buffer durations; P and R vary within an interval of 0.02. The robustness to ID switches increases with increasing buffer duration and decreases with shorter buffer durations. In fact, a larger buffer enables the algorithm to process more data in order to generate more accurate trajectories. We decided to use a buffer size of 50 frames since it provides a good trade-off between tracking latency and performance. The smaller the τ_l, the higher the P. In fact, many false detections are not linked and get discarded because they generate short trajectories. However, R is lower compared to the other cases because true detections are also discarded in the pruning process. A similar behaviour can be spotted when values of $\{\sigma_1, \sigma_2\}$ are small, which is expected since the algorithm does not link short tracks with large spatio-temporal gaps between an end and a start. This is also a constraint for the linkage of short tracks in the case of fast moving targets; small values of $\{\sigma_1, \sigma_2\}$ provides a steep decreasing trend of the likelihood function (Eq. 5.12) and hence low probabilities for linking short tracks. On the other hand, large values of $\{\sigma_1, \sigma_2\}$ lead to a higher R, but a lower P. The last observation is related to the robustness to ID switches as a function of $\{\sigma_1, \sigma_2\}$. Fig. 5.11(c,d) show the poor performance of the algorithm when $\{\sigma_1, \sigma_2\}$ are either too small or too large. On the one hand, with small values of $\{\sigma_1, \sigma_2\}$ each target is likely to have highly fragmented trajectories (multiple identities due to continuous track re-initialisations). On the other hand, larger values of $\{\sigma_1, \sigma_2\}$ provide longer trajectories, but more ID switches between neighbouring targets since the kernel for the linkage is larger (a less steep likelihood function).

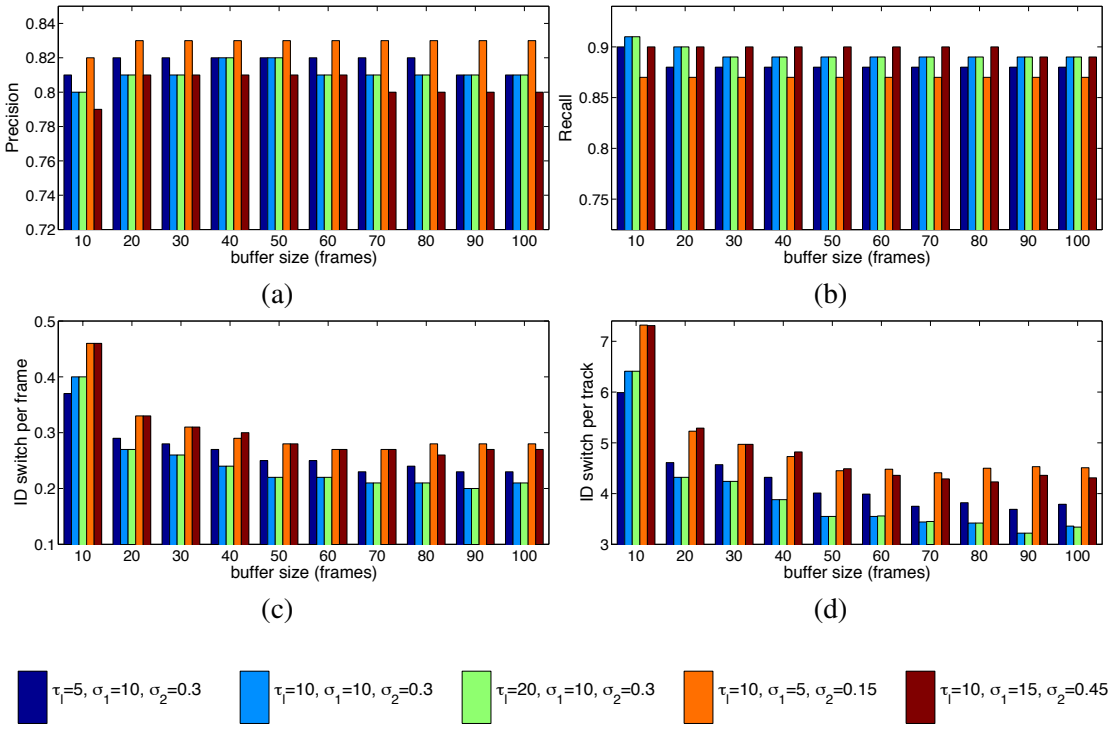


Figure 5.11: Tracking results on B-D1 at varying parameters of GGB. (a) Precision, (b) Recall, (c) Γ and (d) Λ are calculated for different buffer durations (horizontal axis), different τ_l values (dark blue, light blue and green bars) and different $\{\sigma_1, \sigma_2\}$ values (light blue, orange and brown bars). Please see (5.12) for details about the variables.

5.5 Summary

We presented a framework for multi-target target tracking on low-SNR insect datasets. We described a novel detection method to extract features from videos with high-density of targets and a graph-based data association method for multi-target tracking. The detection stage generates candidate target locations relying on gradient information and intensity levels extracted from target-intensity maps. The features are processed using the combination of a method based on Markov Chain Monte Carlo and one based on hierarchical isocontours. Moreover, we used a greedy tracking algorithm that recursively associates detections within a short temporal buffer. The performance of the proposed framework is validated on low-SNR videos and on a publicly available dataset of ants.

Experiments showed that standalone morphological operators [86], threshold-based [51] or template-based algorithms are not accurate enough to detect interacting targets from target-intensity maps. We compared different combinations of MT-TBD with the postprocessing stage presented in Sec. 3.4 and with that proposed in this chapter (GGB). In the case of targets with

indistinguishable appearance and unpredictable motion, a greedy-graph based model based on position features is more reliable than models based on linear motion models (e.g. Kalman filter) and appearance features (e.g. MT-TBD+PP). The combination of MT-TBD+GGB showed better performance than HA+GGB in terms of robustness to ID switches in a scenario with high-density of targets.

Chapter 6

Conclusions

6.1 Summary of methods

In Chapter 3, we presented a Bayesian method for multi-object tracking based on *track-before-detect* (MT-TBD), which utilises Markov Random Fields applied to the particle states to perform tracking of unknown and unlimited number of targets and by managing probabilistically the identity assignment with close objects. To deal with close targets, our approach does not rely on appearance information, but performs a probabilistic optimisation to keep the identity associated to the correct targets. The tracking is performed using a Bayesian recursion via prediction and update steps. The prediction is described with a linear motion model where the unforeseen disturbances are modelled as Gaussian noise. The update is performed using a likelihood function modelled on a particular controlled scenario and used for all our experiments. The particle birth and death at each iteration of the filter is modelled with Markov Random Fields, which assumes the Markovian property. The state estimate of a target is performed via Mean-Shift clustering and supported with Mixture of Gaussians in order to allow an accurate assignment of identities within each single cluster. The sensitivity and experimental analysis were performed on sport and surveillance datasets with variable numbers of moving people, different perspectives, partial and full occlusions, different backgrounds, and moving and static cameras.

In Chapter 4, we presented three measures namely Multiple Extended-target Tracking Error (METE), Multiple-Extended-target Lost-Track Ratio (MELT) and Normalised Identity Changes (NIDC), which quantify accuracy, cardinality, long-term tracking and identity changes for ex-

tended multi-target trackers. These measures are parameter independent, numerically bounded and account for target-size changes. METE provides a holistic error assessment using a trade-off between accuracy and cardinality errors. When trackers have comparable METE values, we showed that accuracy and cardinality error rates can be used separately to analyse more in detail their performance. MELT enables the analysis of tracking performance at varying accuracy levels that can facilitate the selection of trackers for specific applications. NIDC penalises identity changes as a function of the length of the track in which they occur. We compared the new proposed measures with widely used measures for multi-target tracking (Multiple Object Detection Accuracy, Multiple Object Tracking Precision and Identity changes) and we performed an extensive evaluation of MT-TBD comparing its performance with respect to state-of-the-art methods using real-world sequences. MT-TBD showed good flexibility in utilising input coming from different target localisation methods, and by obtaining comparable or superior results with respect to the other methods.

In Chapter 5, we presented a novel feature extraction method and a graph-based detection association algorithm to localise targets in low Signal-to-Noise-Ratio (low-SNR) videos with high density of compact targets (i.e. bees and ant). The detection stage relies on gradient information and intensity levels of target-intensity maps to extract candidate target locations. The features are processed frame-by-frame by using the combination of two methods: one based on Markov Chain Monte Carlo and the other based on hierarchical isocontours. The former exploits the intensity gradient to iteratively align detections on each object. The latter slices the intensity map at different intensity values in order to obtain compact and consistent object shapes through different levels. We then presented a greedy tracking algorithm that recursively associates detections within a short temporal buffer. The performance of the proposed framework is validated on low-SNR videos of bees and on a publicly available dataset of ants. The graph-based tracker is compared with MT-TBD and alternative state-of-the-art tracking methods, such as based on Brownian motion estimation, Kalman filter, energy minimisation via fitting of splines and iterative association based on the Hungarian algorithm.

6.2 Summary of achievements

In this thesis, we addressed three main problems regarding tracking on videos with a high density of targets and the performance evaluation of trackers for extended targets.

The *first problem* involves the implicit assignment and management of target identities within a tracking framework in order to improve the distinction of targets that move close to each other. We showed a situation of partial overlap of targets where the method can reliably keep the identities separate, whereas in the case of full overlap the method was not able to keep track of the occluded target. The track was terminated at the beginning of the occlusion and reinitialised when the target became visible again. In order to improve the performance we used a postprocessing stage that exploits target colour and dynamics to link fragmented (short) tracks into longer ones.

The *second problem* involves the development of three parameter-independent measures capable of evaluating multi-target tracking accuracy, long-term tracking ability and identity changes as a function of the track length. We showed that in comparison with state-of-the-art measures the proposed measures can be used without setting any parameters and that across different applications the evaluation remains consistent. The latter measures also allow one to further analyse the tracking performance in terms of overlap accuracy and cardinality error as two separate pieces of information, as well as lost-track-ratio at different accuracy levels (i.e. overlap values).

The *third problem* involves the formulation of a detection and tracking algorithm that deals with a high density of compact targets. The detector requires the targets all to have the same shape, which is provided as prior information at initialisation. We showed that the proposed detection method achieves better performance than alternatives from the state of the art. However, when targets do not follow the prior shape, the detector may provide incorrect results. An example was observed on the bee dataset, where a bee with opened wings was not detected correctly. We also showed that the greedy-graph based tracking method outperforms state-of-the-art approaches, especially in terms of identity switches. We motivated the choice of using a greedy algorithm over a dynamic programming based one (e.g. Viterbi) since the association is performed on noisy data. In fact, an optimal algorithm would have performed an exhaustive association without accounting for noise.

6.3 Future work

Open challenges in multi-target tracking include the effective extension of feature selection for target-background separability from offline [160] to on-line approaches [148], defining motion models that are flexible to deal with different dynamics of a scene [145], and predicting tracking failures by identifying image regions where trackers are likely to fail [80]. These failures can be

detected by employing interaction models based on track information [83] and potentially solved by strengthening trackers with methods for the self-tuning of parameters [100] (e.g. a resampling strategy for particle filter [142]). Removing the dependence of user interaction is also desirable to make the environment learning stage flexible to context changes [82, 114] and independent from user feedback [109].

The future directions of our work are summarised below:

1. The multi-target track-before-detect presented in Chapter 3 can be improved by including a multi-dynamic switching model [142] to deal with different motions of the observed targets and by developing an automatic method for estimating the filter parameters, such as the noise surrounding the measurement of each point target, the motion model and the likelihood function.
2. The evaluation measures presented in Chapter 4 (METE, MELT and NIDC) can also be applied to other sensing modalities when a 2D target model is used. Given the increasing number of applications employing other target models for 2.5D and 3D tracking [134, 161, 177], future research directions could investigate the extension of the proposed tracking evaluation approaches for these higher-dimensional target models.
3. The detection method for compact objects presented in Sec. 5.2 can be extended by adding a complementary shape-fitting detector [58] or by employing a method for the automatic selection of shape parameters (ellipse) in the case of targets with different sizes.
4. The graph-based association method for tracking presented in Sec. 5.3 can be extended by including appearance and velocity information in the case of targets with potentially different appearance (e.g. people). It can also be extended to real-time applications by performing the association directly on detections (i.e. without precomputing short tracks) by using linear programming.

Finally, there is a growing interest in tracking targets using multiple cameras for increasing the overall field of view [166]. In this case, the target discrimination and identity association techniques need to consider the appearance variability of targets among view-points and across cameras due to changes in illumination, pose and colours.

Bibliography

[1] https://www.mathworks.com/matlabcentral/fileexchange/34040-simple-tracker. Last accessed: December 2013.

[2] http://trecvid.nist.gov/. Last accessed: December 2013.

[3] http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html. Last accessed: December 2013.

[4] http://www.apidis.org/Dataset/. Last accessed: December 2013.

[5] http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html. Last accessed: December 2013.

[6] http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bbenfold_headpose/project.html. Last accessed: December 2013.

[7] http://www.vision.ee.ethz.ch/~aess/iccv2007/. Last accessed: December 2013.

[8] Accuracy (trueness and precision) of measurement methods and results - Part 1: General principles and definitions, Dec. 1994.

[9] R. Ahuja, T. Magnati, and J. Orlin. *Network flow: theory, algorithms, and applications*. Prentice Hall, 2008.

[10] I. Ali and M. N. Dailey. Multiple human tracking in high-density crowds. In *Proc. of Conference on Advanced Concepts for Intelligent Vision Systems*, pages 540–549, Bordeaux, France, Sep. 2009.

[11] S. Ali and M. Shah. A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *Proc. of Computer Vision and Pattern Recognition*, pages 1–6, Minneapolis, MN, USA, Jun. 2007.

[12] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *Proc. of European Conference on Computer Vision*, pages 1–14, Marseille, France, Oct. 2008.

[13] E.L. Andrade, S. Blunsden, and R.B. Fischer. Modelling crowd scenes for event detection. In *Proc. of Internation Conference on Pattern Recognition*, pages 175–178, Hong Kong, China, Sep. 2006.

[14] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43, 2003.

[15] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *Proc. of Computer Vision and Pattern Recognition*, pages 1926–1933, Providence, Rhode Island, USA, Jun. 2012.

[16] N. Anjum and A. Cavallaro. Multi-feature object trajectory clustering for video analysis. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(11):1555–1564, Nov. 2008.

[17] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188, Aug. 2002.

[18] C. Beleznai, B. Fruhstuck, and H. Bischof. Human tracking by fast Mean Shift mode seeking. *Journal of Multimedia*, 1(1):1–8, Apr. 2006.

[19] C. Beleznai and D. Schreiber. Multiple object tracking by hierarchical association of spatio-temporal data. In *Proc. of International Conference on Image Processing*, pages 41–44, Hong Kong, China, Sep. 2010.

[20] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Proc. of Computer Vision and Pattern Recognition*, pages 3457–3464, Colorado Springs, USA, Jun. 2011.

[21] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using K-Shortest paths optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, Sep. 2011.

[22] P. Berens. CircStat: A Matlab toolbox for circular statistics. *Journal of Statistical Software*, 31(10):1–21, Sep. 2009.

[23] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society Series B Methodological*, 48(3):259–302, Mar. 1986.

[24] J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. In *Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 125–132, Nice, France, Oct. 2003.

[25] S.S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, Jan. 2004.

[26] S. Blunsden, E. Andrade, and R. Fisher. Non parametric classification of human interaction. In *Proc. of Iberian Conference on Pattern Recognition and Image Analysis*, pages 347–354, Girona, Spain, Jun. 2007.

[27] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833, Sep. 2011.

[28] L. M. Brown, A. W. Senior, Y.-L. Tian, J. Connell, A. Hampapur, C.-F. Shu, H. Merkl, and M. Lu. Performance evaluation of surveillance systems under varying conditions. In *Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 1–8, Breckenridge, Colorado, USA, Jan. 2005.

[29] A.A. Butt and R.T. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *Proc. of Computer Vision and Pattern Recognition*, pages 1846–1853, Portland, OR, USA, Jun. 2013.

[30] S. Buzzi, M. Lops, L. Venturino, and M. Ferri. Track-before-detect procedures in a multitarget environment. *IEEE Trans. of Aerospace and Electronic Systems*, 44(3):1135–1150, Jul. 2008.

[31] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proc. of European Conference on Computer Vision*, pages 778–792, Crete, Greece, Sep. 2010.

[32] D. Chau, F. Bremond, and M. Thonnat. Online evaluation of tracking algorithm performance. In *Proc. of Crime Detection and Prevention*, pages 1–6, London, UK, Dec. 2009.

[33] M.-Y. Chen, H. Li, and A. Hauptmann. Informedia @ trecvid 2009: Analyzing video motions. In *TRECVID Workshop at NIST*, Gaithersburg, MD, Nov. 2009.

[34] Z. Chen. Bayesian filtering: from Kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, Jan. 2003.

[35] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, Oct. 2005.

[36] R.T. Collins. Multitarget data association with higher-order motion models. In *Proc. of*

Computer Vision and Pattern Recognition, pages 1744–1751, Providence, Rhode Island, USA, Jun. 2012.

[37] R.T. Collins, A.J. Lipton, T. Kanade, H. Fujiyoshi, and D. Duggins et al. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, The Robotics Institute, Carnegie Mellon University, Pittsburgh PA, 2000.

[38] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

[39] Dorin Comaniciu and Peter Meer. Distribution free decomposition of multivariate data. *Pattern Analysis and Applications*, 2(1):22–30, Apr. 1999.

[40] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, Cambridge, MA, 1990.

[41] N. Courty and T. Corpetti. Crowd motion capture. *Computer Animation and Virtual Worlds*, 18(4-5):361–370, Sep. 2007.

[42] J.C. Crocker and D.G. Grier. Methods of digital video microscopy for colloidal studies. *Journal of Colloid and Interface Science*, 179(1):298–310, Apr. 1996.

[43] J. Czyz, B. Ristic, and B. Macq. A particle filter for joint detection and tracking of color objects. *Image and Vision Computing*, 25:1271–1281, Aug. 2007.

[44] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of Computer Vision and Pattern Recognition*, pages 886–893, San Diego, CA, USA, Jun. 2005.

[45] F. Daniyal and A. Cavallaro. Abnormal motion detection in crowded scenes using local spatio-temporal analysis. In *Proc. of Conference on Acoustics, Speech and Signal Processing*, pages 1944–1947, Prague, Czech Republic, May 2011.

[46] D. Delannay, N. Danhier, and C. De Vleeschouwer. Detection and recognition of sports(wo)men from multiple views. In *Proc. of Conference on Distributed Computing Systems*, pages 1–7, Como, Italy, Sep. 2009.

[47] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: an evaluation of the state of the art. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(4):743–761, Apr. 2012.

[48] A. Doucet, J.F.G. de Freitas, and N.J. Gordon. *Sequential Monte Carlo methods in practice*. Springer-Verlang, 2001.

[49] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, Singapore, 2001.

[50] R. Eshel and Y. Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Proc. of Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, US, Jun. 2008.

[51] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sep. 2010.

[52] A.F. Frangi, W.J. Niessen, K.L. Vinken, and M.A. Viergever. Multiscale vessel enhancement filtering. In *Proc. of Medial Image Computing and Computer-Assisted Intervention*, pages 130–137, Cambridge, MA, USA, Oct. 1998.

[53] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4(6):933–969, Jan. 2003.

[54] B. E. Fridling and O. E. Drummond. Performance evaluation methods for multiple-target-tracking algorithms. In *Proc. of Signal Data Processing of Small Targets*, volume 1481, pages 371–383, Orlando, FL, USA, Oct. 1991.

[55] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(11):2188–2202, Nov. 2011.

[56] B. Georgescu, I. Shimshoni, and P. Meer. Mean-Shift based clustering in high dimensions: a texture classification example. In *Proc. of International Conference on Computer Vision*, pages 456–463, Beijing, China, 2003, Oct. 2003.

[57] A.V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithms. *Journal of Algorithms*, 22(1):1–29, Jan. 1997.

[58] R.C. Gonzalez and R.E. Woods. *Digital image processing*. Pearson Prentice Hall, Pearson Education, Inc., 2008.

[59] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[60] D.-C. He and L. Wang. Texture unit, texture spectrum, and texture analysis. *IEEE Trans. on Geoscience and Remote Sensing*, 28(4):509–512, Jul. 1990.

[61] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4):657–662, Apr. 2006.

[62] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, May 1995.

[63] D. Helbing, P. Molnar, I.J. Farkas, and K. Bolay. Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design*, 28(3):361–383, Mar. 2001.

[64] D. Herman. Multi-object tracking algorithm for biological motion using Kalman filter and Munkres algorithm, http://studentdavestutorials.weebly.com/multi-bugobject-tracking.html. Last accessed: December 2013.

[65] R. Hess and A. Fern. Discriminatively trained particle filters for complex multi-object tracking. In *Proc. of Computer Vision and Pattern Recognition*, pages 240–247, Miami, FL, USA, Jun. 2009.

[66] J. R. Hoffman and R. P. S. Mahler. Multitarget miss distance via optimal assignment. *IEEE Trans. on Systems, Man and Cybernetics - Part A*, 34(3):327 – 336, May 2004.

[67] R. Hogg, J. McKean, and A. Craig. *Introduction to Mathematical Statistics*. Upper Saddle River, NJ, 2005.

[68] R. Hoseinnezhad, B.-N. Vo, D. Suter, and B.-T. Vo. Multi-object filtering from image sequence without detection. In *Proc. of Conference on Acoustics, Speech and Signal Processing*, pages 1154–1157, Dallas, Texas, USA, Mar. 2010.

[69] J.R.M. Hosking and J.R. Wallis. Parameter and quantile estimation for the generalized Pareto distribution. *Technometrics*, 29(3):339–349, Aug. 1987.

[70] M. Hu, S. Ali, and M. Shah. Learning motion patterns in crowded scenes using motion flow field. In *Proc. of International Conference on Pattern Recognition*, pages 1–5, Tampa, FL, USA, Dec. 2008.

[71] M.-C. Hu, M.-H. Chang, J.-L Wu, and L. Chi. Robust camera calibration and player tracking in broadcast basketball video. *IEEE Trans. on Multimedia*, 13(2):266–279, Mar. 2011.

[72] W. Hu, X. Xiao, Z. Fu, D. Xie T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, Sep. 2006.

[73] C. Huang, Y. Li, and R. Nevatia. Multiple target tracking by learning based hierarchical association of detection responses. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(4):898–910, Apr. 2013.

[74] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proc. of European Conference on Computer Vision*, pages 788–801, Marseille, France, Oct. 2008.

[75] C. Hue, J.-P. Le Cadre, and P. Perez. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Trans. on Signal Processing*, 50(2):309–325, Feb. 2002.

[76] R.L. Hughes. The flow of human crowds. *Annual Review of Fluid Mechanics*, 35(35):169–182, Jan. 2003.

[77] H. Idrees, N. Warner, and M. Shah. Tracking in dense crowds using prominence and neighborhood motion concurrence. *Image and Vision Computing*, 32(1):14–26, Jan. 2014.

[78] M. Isard and A. Blake. CONDENSATION - Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, Aug. 1998.

[79] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: automatic detection of tracking failures. In *Proc. of International Conference on Pattern Recognition*, pages 2756–2759, Istambul, Turkey, Aug. 2010.

[80] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, Jul. 2012.

[81] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(2):319–336, Feb. 2009.

[82] A. Kembhavi, T. Yeh, and L. S. Davis. Why did the people cross the road (there)? Scene understanding using probabilistic logic models and common sense reasoning. In *Proc. of European Conference on Computer Vision*, pages 693–706, Crete, Greece, Sep. 2010.

[83] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, Nov. 2005.

[84] Z. Khan, T. Balch, and F. Dellaert. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *Trans. on Pattern Analysis and Machine Intelligence*, 28(12):1960–1972, Dec. 2006.

[85] K. Kim, M. Grundmann, A. Shamir, I. Matthews, J. Hodgins, and I. Essa. Motion field to predict play evolution in dynamic sport scenes. In *Proc. of Computer Vision and Pattern Recognition*, pages 840–847, San Francisco, CA, USA, Jun. 2010.

[86] Y. Kimori, N. Baba, and N. Morone. Extended morphological processing: a practical method for automatic spot detection of biological markers from microscopic images. *BMC Bioinformatics*, 11(373):1–13, Jul. 2010.

[87] R. Kindermann and J. L. Snell. *Markov Random Fields and their applications*. American Mathematical Society, Providence, Rhode Island, 2000.

[88] N.S. Kopeika. *A system engineering approach to imaging*. SPIE press, 1998.

[89] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages 1446–1453, Miami, FL, USA, Jun. 2009.

[90] L. Kratz and K. Nishino. Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(5):987–1002, May 2012.

[91] D.-J. Kroon. *Segmentation of the mandibular canal in cone-beam CT data*. PhD thesis, University of Twente, 2006.

[92] H.W. Kuhn. The Hungarian method for the assignement problem. *Naval Research Logistics Quarterly*, 2:843–854, 1955.

[93] C.H. Kuo, C. Huang, and R. Nevatia. Inter-camera association of multi-target tracks by on-line learned appearance affinity models. In *Proc. of European Conference on Computer Vision*, pages 383–396, Crete, Greece, Sep. 2010.

[94] C.H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *Proc. of Computer Vision and Pattern Recognition*, pages 685–692, San Francisco, CA, USA, Jun. 2010.

[95] C.H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking?

In *Proc. of Computer Vision and Pattern Recognition*, pages 1217–1224, Colorado Springs, USA, Jun. 2011.

[96] T. De Laet, H. Bruyninckx, and J. De Schutter. Shape-based online multitarget tracking and detection for targets causing multiple measurements: Variational Bayesian clustering and lossless data association. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(12):2477–2491, Dec. 2011.

[97] B. Leibe, Konrad Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(10):1683–1698, Oct. 2008.

[98] H. Li, L. Bao, Z. Gao, A. Overwijk, W. Liu, L.-F. Zhang, S.-I Yu, M.-Y. Chen, F. Metze, and A. Hauptmann. Informedia @ trecvid 2010. In *TRECVID Workshop at NIST*, Gaithersburg, MD, Nov. 2010.

[99] H. Li, C. Shen, and Q. Shi. Real-time visual tracking using compressive sensing. In *Proc. of Computer Vision and Pattern Recognition*, pages 1305–1312, Providence, RI, USA, Jun. 2011.

[100] M. Li, T. Tan, W. Chen, and K. Huang. Efficient object tracking by incremental self-tuning particle filtering on the affine groups. *IEEE Trans. on Image Processing*, 21(3):1298–1313, Sep. 2012.

[101] M. Li, Z. Zhang, K. Huang, and T. Tan. Rapid and robust human detection and tracking based on omega-shape features. In *Proc. of International Conference on Image Processing*, pages 2545–2548, Cairo, Egypt, November 2009.

[102] R. Li and R. Chellappa. Group motion segmentation using a spatio-temporal driving force model. In *Proc. of Computer Vision and Pattern Recognition*, pages 2038–2045, San Francisco, CA, USA, Jun. 2010.

[103] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Proc. of Computer Vision and Pattern Recognition*, pages 2953–2960, Miami, FL, USA, 2009, 20-25 Jun. 2009.

[104] Y. Li and R. Nevatia. Key object driven multi-category object recognition, localization and tracking using spatio-temporal context. In *Proc. of European Conference on Computer Vision*, pages 409–422, Marseille, France, Oct. 2008.

[105] K. Lia, E.D. Millera, M. Chenb, T. Kanadea, L.E. Weissa, and P.G. Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Medical Image Analysis*, 12(5):546–566, Oct. 2008.

[106] A.-A Liu and Z. Gao. Trecvid 2010 surveillance event detection by mmm-tju. In *TRECVID Workshop at NIST*, Gaithersburg, MD, Nov. 2010.

[107] J. Liu, P. Carr, R.T. Collins, and Y. Liu. Tracking sports players with context-conditioned motion models. In *Proc. of Computer Vision and Pattern Recognition*, pages 1830–1837, Portland, OR, USA, Jun. 2013.

[108] D.G. Lowe. Object recognition from local scale-inveriant feature. In *Proc. of International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, Sep. 1999.

[109] E. Maggio and A. Cavallaro. Learning scene context for multiple object tracking. *IEEE Trans. on Image Processing*, 18(8):1873–1884, Aug. 2009.

[110] E. Maggio and A. Cavallaro. *Video tracking: theory and practice*. Wiley, 2011.

[111] E. Maggio, F. Smeraldi, and A. Cavallaro. Adaptive multi-feature tracking in a particle filtering framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(10):1348–1359, Oct. 2007.

[112] R. Mahler. *Random-set approach to data fusion*. SPIE, 1994.

[113] R. Mahler. A theoretical foundation for the Stein-Winter Probability Hypothesis Density (PHD) multitarget tracking approach. In *Proc. of MSS National Symposium on Sensor and Data Fusion*, San Antonio, Texas, USA, 2002.

[114] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Trans. on Systems, Man and Cybernetics - Part B*, 35(3):397–408, Jun. 2005.

[115] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, pages 384–396, Cardiff, UK, Sep. 2002.

[116] MATLAB. *vesion 8.2.0 (2013b)*. The MathWorks Inc., Natick, Massachusetts, 2013.

[117] R. Mazzon, S. F. Tahir, and A. Cavallaro. Person re-identification in crowd. *Pattern Recognition Letters*, 33(15):1828–1837, Oct. 2012.

[118] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using Social Force Model. In *Proc. of Computer Vision and Pattern Recognition*, pages 935–942, Miami, FL, USA, Jun. 2009.

[119] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *Proc. of International Conference on Computer Vision*, pages 1792–1799, Beijing, China, Oct. 2005.

[120] Y. Mingqiang, K. Kidiyo, and R. Joseph. A survey of shape feature extraction techniques. *Pattern Recognition*, pages 43–90, Jul. Peng-Yeng Yin (Ed.), 2008.

[121] A. Mitiche and P. Bouthemy. Computation and analysis of image motion: A synopsis of current problems and methods. *International Journal of Computer Vision*, 19(1):29–55, Jul. 1996.

[122] J. Munkres. Algorithms for assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, Mar. 1957.

[123] T. Nawaz and A. Cavallaro. PFT: a protocol for evaluating video trackers. In *Proc. of International Conference on Image Processing*, pages 2325–2328, Brussels, Belgium, Sep. 2011.

[124] T. Nawaz and A. Cavallaro. A protocol for evaluating video trackers under real-world conditions. *IEEE Trans. on Image Processing*, 22(4):1354–1361, Apr. 2013.

[125] W. Ng, J. Li, S. Godsill, and J. Vermaak. A review of recent results in multiple target tracking. In *Proc. of Image and Signal Processing and Analysis*, pages 40–45, Trieste, Italy, Sep. 2005.

[126] S. Oh and S. Sastry. Tracking on a graph. In *Proc. of Symposium on Information Processing in Sensor Networks*, pages 195–202, Los Angeles, CA, USA, Apr. 2005.

[127] S. M. Oh, J.M. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, 77(1-3):103–124, May 2008.

[128] K. Okuma, A. Talenghani, N. De Freitas, J.J. Little, and D.G. Lowe. A boosted particle filter: multitarget detection and tracking. In *Proc. of European Conference on Computer Vision*, pages 28–39, Prague, Czech Republic, May 2004.

[129] N.M. Oliver, B. Rosario, and A.P. Pentland. A Bayesian computer vision system for modeling human interaction. *IEEE Trans. of Pattern Analysis and Machine Intelligence*, 22(8):831–843, Aug. 2000.

[130] A. Papoulis and S.U. Pillai. *Probability, random variables and stochastic processes*. Mc Graw Hill, 2002.

[131] M. Piccardi. Background subtraction techniques: a review. In *Proc. of International Conference on Systems, Man and Cybernetics*, pages 3099–3104, The Hauge, Netherlands, Oct. 2004.

[132] H. Pirsiavash, D. Ramanan, and C.C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proc. of Computer Vision and Pattern Recognition*, pages 1201–1208, Providence, RI, USA, Jun. 2011.

[133] F. Poiesi, F. Danyial, and A. Cavallaro. Detector-less ball localisation using context and motion flow analysis. In *Proc. of International Conference on Image Processing*, pages 3913–3916, Hong Kong, China, Sep. 2010.

[134] D. Poullin and M. Flecheux. Passive 3D tracking of low altitude targets using DVB (SFN Broadcasters). *Aerospace and Electronic Systems Magazine*, 27(11):36–41, Nov. 2012.

[135] Z. Qin and C.R. Shelton. Improving multi-target tracking via social grouping. In *Proc. of Computer Vision and Pattern Recognition*, pages 1972–1978, Providence, RI, USA, Jun. 2012.

[136] L.R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceeding of the IEEE*, 77(0018-9219):257–286, Feb. 1989.

[137] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Trans. on Image Processing*, 14(3):294–307, Mar. 2005.

[138] S.H. Razatofighi, R. Hartley, and W.E. Hughes. A new approach for spot detection in total internal reflection fluorescence microscopy. In *International Symposium on Biomedical Imaging*, pages 860–863, Barcelona, Spain, May 2012.

[139] D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 24(6):843–854, Jan. 1979.

[140] S.H. Rezatofighi, S. Gould, R. Hartley, K. Mele, and W.E. Hughes. Application of the IMM-JPDA filter to multiple target tracking in total internal reflection fluorescence mi-

croscopy images. In *Proc. of Medical Image Computing and Computer-Assisted Intervention*, pages 357–364, Nice, France, Oct. 2012.

[141] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

[142] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, 2004.

[143] B. Ristic, B.-N. Vo, D. Clark, and B.-T. Vo. A metric for performance evaluation of multi-target tracking algorithms. *IEEE Trans. on Signal Processing*, 59(7):3452–3457, Jul. 2011.

[144] M. Rodriguez, S. Ali, and T. Kanade. Tracking in unstructured crowded scenes. In *Proc. of International Conference on Computer Vision*, pages 1389–1396, Kyoto, Japan, Sep. 2009.

[145] M. Rodriguez, I. Laptev, J. Sivic, and JY. Audibert. Density-aware person detection and tracking in crowds. In *Proc. of International Conference on Computer Vision*, pages 2423–2430, Barcelona, Spain, Nov. 2011.

[146] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert. Data-driven crowd analysis in videos. In *Proc. of International Conference on Computer Vision*, pages 1235–1242, Barcelona, Spain, Nov. 2011.

[147] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. In *Proc. of Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA, 2007 2007.

[148] S. Salti, A. Cavallaro, and L. Di Stefano. Adaptive appearance modeling for video tracking: survey and evaluation. *IEEE Trans. of Image Processing*, 21(10):4334–4348, Oct. 2012.

[149] J.C. SanMiguel, A. Cavallaro, and J.M. Martinez. Adaptive on-line performance evaluation of video trackers. *IEEE Trans. on Image Processing*, 21(5):2812–2823, May 2012.

[150] K. Sato and J.K. Aggarwal. Temporal spatio-velocity transform and its application to tracking and interaction. *Computer Vision and Image Understanding*, 96:100–128, Sept. 2004.

[151] S. Saxena, F. Bremond, M. Thonnat, and R. Ma. Crowd behavior recognition for video surveillance. In *Proc. of International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 970–981, Juan-les-Pins, France, Oct. 2008.

[152] R.E. Schapire and Y. Singer. *Improved Boosting Algorithms Using Confidence-rated Predictions*. Machine Learning, 1999.

[153] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. on Signal Processing*, 56(8):3447–3457, Aug. 2008.

[154] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. In *Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Hawaii, USA, Dec. 2001.

[155] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(1):51–65, Jan. 2005.

[156] I. Smal, M. Loog, and E. Meijering. Quantitative comparison of spot detection methods in fluorescence microscopy. *IEEE Trans. on Medical Imaging*, 29(2):282–301, Feb. 2010.

[157] K. Smith, S.O. Ba, J.-M. Odobez, and D. Gatica-Perez. Tracking the visual focus of attention for a varying number of wandering people. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(7):1–17, Jul. 2008.

[158] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, 1999, pp. 173-174.

[159] S. Stalder, H. Grabner, and L. Van Gool. Cascaded confidence filtering for improved tracking-by-detection. In *Proc. of European Conference on Computer Vision*, pages 369–382, Crete, Greece, 2010, Sep. 2010.

[160] B. Stenger, T. Woodley, and R. Cipolla. Learning to track with multiple observers. In *Proc. of Computer Vision and Pattern Recognition*, pages 2647–2654, Washington, DC, USA, Jun. 2009.

[161] X. Suau, J. Ruiz-Hidalgo, and J. R. Casas. Real-time head and hand tracking based on 2.5D data. *IEEE Trans. on Multimedia*, 14(3):575–585, Jun. 2012.

[162] H.-I. Suk, A.K. Jain, and S.-W. Lee. A network of dynamic probabilistic models for human interaction analysis. *IEEE Trans. on Circuits and Systems for Video Technology*, 21(7):932–945, Jul. 2011.

[163] D. Sun, S. Roth, and M.J. Black. Secrets of optical flow estimation and their principles. In *Proc. of Computer Vision and Pattern Recognition*, pages 2432–2439, San Francisco, CA, USA, Jun. 2010.

[164] G. Sundaramoorthi, A. Mennucci, S. Soatto, and A. Yezzi. Tracking deforming objects by filtering and prediction in the space of curves. In *Proc. of Decision and Control*, pages 2395–2401, Shanghai, China, Dec. 2009.

[165] M. Taj and A. Cavallaro. *Recognizing interactions in video*, volume 282/2010. Intellig. Multimed. Analys. for Secur. Applic, Springer, 2010.

[166] M. Taj and A. Cavallaro. Distributed and decentralized multi-camera tracking. *IEEE Signal Processing Magazine*, 28(3):46–58, May 2011.

[167] Murtaza Taj. *Tracking Interacting Target in Multi-modal Sensors*. PhD thesis, The School of Electronic Engineering and Computer Science, Queen Mary University of London, 2009.

[168] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Apr. 1991.

[169] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(10):1–15, Oct. 2008.

[170] A. Veeraraghavan, R. Chellappa, and M. Srinivasan. Shape-and-Behavior-Encoded tracking of bee dances. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(3):463–476, Mar. 2008.

[171] J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *Proc. of International Conference on Computer Vision*, pages 1110–1116, Nice, France, Oct. 2003.

[172] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, Jul. 2005.

[173] B.-Ngu Vo, B.-Tuong Vo, N.-T. Pham, and D. Suter. Joint detection and estimation of multiple objects from image observations. *IEEE Trans. on Signal Processing*, 58(10):5129–5241, Oct. 2010.

[174] N. von Hoyningen-Huene and M. Beetz. Robust real-time multiple target tracking. In *Proc. of Asian Conference on Computer Vision*, pages 247–256, Xi'an, China, Sep. 2009.

[175] A. Waibel, T. Schultz, M. Bett, M. Denecke, R. Malkin, I. Rogina, and R. Stiefelhagen. SMaRT: the Smart Meeting Room Task at ISL. In *Proc. of Conference on Acoustics, Speech and Signal Processing*, pages 752–755, Hong Kong, China, Apr. 2003.

[176] X. Wang, X. Ma, and W.E.L. Grimson. Unsupervised activity perception in crowded and complicated scenes using Hierarchical Bayesian model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(3):539–555, Mar. 2009.

[177] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. Monocular visual scene understanding: Understanding multi-object traffic scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(4):882–897, Apr. 2013.

[178] J.K. Wolf, A.M. Viterbi, and G.S. Dixon. Finding the best set of k paths through a trellis with application to multitarget tracking. *IEEE Trans. on Aerospace and Electronic Systems*, 25(2):287–296, Mar. 1989.

[179] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, Nov. 2007.

[180] H. Wu, A. C. Sankaranarayanan, and R. Chellappa. Online empirical evaluation of tracking algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1443–1458, Aug. 2010.

[181] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *Proc. of Computer Vision and Pattern Recognition*, pages 2034–2041, Providence, RI, USA, Jun. 2012.

[182] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: a review. *Neurocomputing*, 74:3823–3831, Aug. 2011.

[183] L. Yang, Z. Qiu, G. A.H. Greenaway, and W. Lu. A new framework for particle detection in low-SNR fluorescence live-cell images and its application for improved particle tracking. *IEEE Trans. on Biomedical Engineering*, 59(7):2040–2050, Jul. 2012.

[184] M. Yang, F. Lv, W. Xu, and Y. Gong. Detection driven adaptive multi-cue integration for multiple human tracking. In *Proc. of International Conference on Computer Vision*, pages 1554–1561, Kyoto, Japan, Sep. 2009.

[185] A. Yao, J. Gall, and L.V. Gool. A Hough Transform-based voting framework for action recognition. In *Proc. of Computer Vision and Pattern Recognition*, pages 2061–2068, San Francisco, CA, USA, Jun. 2010.

[186] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Journal ACM Computing Surveys*, 38(4):1–45, Dec. 2006.

[187] F. Yin, D. Makris, and S. A. Velastin. Performance evaluation of object tracking algorithms. In *Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Rio de Janeiro, Brazil, Oct. 2007.

[188] Q. Yu and G. Medioni. Motion pattern interpretation and detection for tracking moving vehicles in airborne videos. In *Proc. of Computer Vision and Pattern Recognition*, pages 2671–2678, Miami, FL, USA, Jun. 2009.

[189] Z. Zhang, H. Gunes, and M. Piccardi. An accurate algorithm for head detection based on XYZ and HSV hair and skin color models. In *Proc. of International Conference on Image Processing*, pages 1644–1647, San Diego, CA, USA, Oct. 2008.

[190] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environments. In *Proc. of Computer Vision and Pattern Recognition*, pages 406–413, Washington, DC, USA, Jul. 2004.

[191] B. Zhou, X. Tang, and X. Wang. Measuring crowd collectiveness. In *Proc. of Computer Vision and Pattern Recognition*, pages 3049–3056, Portland, OR, USA, Jun. 2013.

[192] B. Zhou, X. Wang, and X. Tang. Random field topic model for semantic region analysis in crowded scenes from tracklets. In *Proc. of Computer Vision and Pattern Recognition*, pages 3441–3448, Colorado Springs, Colorado, USA, Jun. 2011.

[193] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Proc. of Computer Vision and Pattern Recognition*, pages 2871–2878, Providence, RI, USA, Jun. 2012.